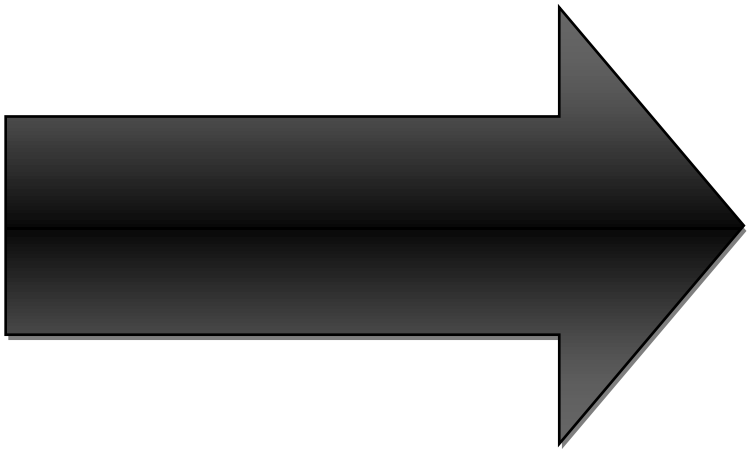
File Version Update: (Dated: 21-May-2011)

This version of file contains:

- Content of the Course (Done)
- FAQ updated version.(These must be read once because some very basic definition and question are being answered)
- Glossary updated version.(These must be read once because some very basic terms are being explained which you even might not found in the book) (Done)
- Solved Past Assignment Selected for MID Term. (Done)
- Solved Question From Mid-Term Papers (Done))
- Current Papers (spring-2011)
- MCQs GIGA File (with references) (Done)

Note: Use Table Of Content to view the Topics, In PDF(*Portable Document Format*) format , you can check Bookmarks menu.



Disclaimer: There might be some human errors, if you find please let me know at pak.nchd@gmail.com , duplication of data may be possible but at least possible level.

----:Table of Content:----

Table of Content

==========>

Introduction To OOP

Course Content:

Introduction, Abstraction & Inheritance, Introduction to Generalization & Specialization, Multiple Inheritance & Associations, Object-Oriented Modeling, Introduction to Objects and Classes, Constructors, Destructor, Accessor Functions & this Pointer, Constant data members & Static Variables, Constant data members & Static Variables& Array of objects, new Operator & Getter and Setter, Composition, Composition& Aggregation and Friend Functions, Operator overloading, Inheritance, Access Specifiers, Copy Constructor &Assignment Operator, Overriding, Types of Inheritance, Polymorphism, Abstract & Concrete Classes, Polymorphism – Case Study, Multiple Inheritance, Generic Programming, Templates and Friends, Generic Algorithms Revisited, Cursors & Vectors, Standard Template Library, Iterators, Techniques for Error Handling, Exception Handling.

FAQ updated version.

Question: What is Abstraction?

Answer: The importance of abstraction is derived from its ability to hide irrelevant details and from the use of names to reference objects. Abstraction is essential in the construction of programs. It places the emphasis on what an object is or does rather than how it is represented or how it works. Thus, it is the primary means of managing complexity in large programs.

Question: What is a Class Diagram?

Answer: A class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects.

Question: What is Method Overriding?

Answer: Method overriding is a language feature that allows a subclass to override a specific implementation of a method that is already provided by one of its super-classes. A subclass can give its own definition of methods but need to have the same signature as the method in its super-class. This means that when overriding a method the subclass's method has to have the same name and parameter list as the super-class's overridden method.

Question: What is Operator Overloading?

Answer: The operator overloading is a specific case of polymorphisms in which some or all of operators like +, - or == are treated as polymorphic (multi) functions and as such have different behaviors depending on the types of its arguments.

Question: What is Method Overloading?

Answer: The method overloading is the ability to define several methods (in same class) all with the same name but different on the basis of
i) number of parameters ii) types of parameters.

Question: What is Polymorphisms?

Answer: Polymorphism is a generic term that means 'many shapes'. More precisely Polymorphism means the ability to request that the same operations be performed by a wide range of different types of things.

Question: What is Inheritance?

Answer: Ability of a new class to be created, from an existing class by extending it, is called inheritance.

Question: What is a base class?

Answer: When inheritance is used to create a new class from another, the new class is called the subclass or derived class, and the class from which it was derived is called the base class.

Question: What is a concrete class?

Answer: A concrete class is one that can be used to directly create, or instantiate objects, unlike an abstract base class which can only be used as a base class for other classes which eventually lead to concrete classes

Question: What are data members?

Answer: Objects are miniature programs, consisting of both code and data. The code consists of a series of member functions. The data items are called data members.

Question: What is a constructor?

Answer: Objects are complete, miniature programs and, like any good programs, have well defined initialization and termination phases. They have special routines (i.e. member functions) to look after this. The initialization routine is called the constructor, and C++ ensures that every object is properly initialized by calling its constructor. The designer of the object can have more than one constructor, a situation called overloading and then the compiler will select between them depending on exactly what arguments are passed to the constructor function. However there must always be a default constructor, to be used when no information is supplied.

Question: What is a destructor?

Answer: The termination routine is called the destructor, and C++ will provide a default if none is supplied. If, during the lifetime of the object, it uses heap memory then the designer of the object must provide a destructor function to release such memory to avoid a memory leak.

Question: What is global variable?

Answer: Global variables can be accessed throughout a program. Another way to put this is to say they have global scope.

Question: What is local variable?

Answer: Local variables can only be accessed within the function, or more specifically the compound statement in which they are declared. Another way to put this is to say they have local scope.

Question: What is a null pointer?

Answer: A null pointer is a pointer that is currently pointing to nothing. Often pointers are set to zero to make them null pointers or tested against zero to see if they are null or not.

Question: What is a pointer?

Answer: A pointer is a variable that holds the address of another variable or object.

Question: What is meant by protected?

Answer: The protected keyword in the class statement means that the following members of the class are not available to users of the objects of the class, but can be used by any subclass that inherits from it, and consequently forms part of its implementation.

Question: What is OOP?

Answer: The object oriented programming is commonly known as OOP. Most of the languages are developed using OOP concept. Object-oriented programming (OOP) is a programming concept that uses "objects" to develop a system. An object hides the implementation details and exposes only the functionalities and parameters it requires to its client. Here also an object shares the same concept as that of a bike. While driving a motor bike, we are unaware of its implementation details such as how it is developed, internal working of gears etc.? We know only the functions or actions it can perform.

Question: What are the various elements of OOP?

Answer: Various elements of OOP are:

- Object
- Class Method
- Encapsulation
- Information Hiding
- Inheritance
- Polymorphism

Question: What are the characteristics of Object Oriented programming language?

Answer: Some key features of the Object Oriented programming are: Emphasis on data rather than procedure Programs are divided into entities known as objects Data Structures are designed such that they characterize objects Functions that operate on data of an object are tied together in data structures Data is hidden and cannot be accessed by external functions Objects communicate with each other through functions New data and functions can be easily added whenever necessary Follows bottom up design in program design

Question: What are the basic Concepts used in the Object-Oriented Programming language?

Answer: Object Class Data Abstraction and Encapsulation Polymorphism Inheritance Message passing Dynamic binding

Question: What Is An Object? (Object-Oriented Technology)

Answer: There are many definitions of an object, such as found in [Booch 91, p77]: "An object has state, behavior, and identity; the structure and behavior of similar objects are defined in their common class; the terms instance and object are interchangeable". This is a "classical languages" definition, as defined in [Coplien 92, p280], where "classes play a central role in the object model", since they do not in prototyping/delegation languages. "The term object was first formally applied in the Simula language, and objects typically existed in Simula programs to simulate some aspect of reality" [Booch 91, p77]. Other definitions referenced by Booch include Smith and Tockey: "an object represents an individual, identifiable item, unit, or entity, either real or abstract, with a well-defined role in the problem domain." and [Cox 91]: "anything with a crisply defined boundary" (in context, this is "outside the computer domain". A more conventional definition appears on pg 54). Booch goes on to describe these definitions in depth. [Martin 92, p 241] defines: "An "object" is anything to which a concept applies", and "A

concept is an idea or notion we share that applies to certain objects in our awareness". [Rumbaugh 91] defines: "We define an object as a concept, abstraction or thing with crisp boundaries and meaning for the problem at hand." [Shlaer 88, p 14] defines: "An object is an abstraction of a set of real-world things such that:

Question: What Is Object Encapsulation (Or Protection)?

Answer: [Booch 91, p. 45] defines: "Encapsulation is the process of hiding all of the details of an object that do not contribute to its essential characteristics." [Coad 91, 1.1.2] defines: "Encapsulation (Information Hiding). A principle, used when developing an overall program structure, that each component of a program should encapsulate or hide a single design decision... The interface to each module is defined in such a way as to reveal as little as possible about its inner workings. [Oxford, 1986]" Some languages permit arbitrary access to objects and allow methods to be defined outside of a class as in conventional programming. Simula and Object Pascal provide no protection for objects, meaning instance variables may be accessed wherever visible. CLOS and Ada allow methods to be defined outside of a class, providing functions and procedures. While both CLOS and Ada have packages for encapsulation, CLOS's are optional while Ada's methodology clearly specifies class-like encapsulation (Adts). However most object-oriented languages provide a well defined interface to their objects thru classes. C++ has a very general encapsulation/protection mechanism with public, private and protected members. Public members (member data and member functions) may be accessed from anywhere. A Stack's Push and Pop methods will be public. Private members are only accessible from within a class. A Stack's representation, such as a list or array, will usually be private. Protected members are accessible from within a class and also from within subclasses (also called derived classes). A Stack's representation could be declared protected allowing subclass access. C++ also allows a class to specify friends (other (sub)classes and functions), that can access all members (its representation). Eiffel 3.0 allows exporting access to specific classes.

Question: What Is A Class?

Answer: A class is a general term denoting classification and also has a new meaning in object-oriented methods. Within the OO context, a class is a specification of structure (instance variables), behavior (methods), and inheritance (parents, or recursive structure and behavior) for objects. As pointed out above, classes can also specify access permissions for clients and derived classes, visibility and member lookup resolution. This is a feature-based or intensional definition, emphasizing a class as a descriptor/constructor of objects (as opposed to a collection of objects, as with the more classical extensional view, which may begin the analysis process). Original Aristotlean classification defines a "class" as a generalization of objects: [Booch 91, p93] "a group, set, or kind marked by common attributes or a common attribute; a group division, distinction, or rating based on quality, degree of competence, or condition".

Question: What Is A Meta-Class?

Answer: Meta-Class is a class' class. If a class is an object, then that object must have a class (in classical OO anyway). Compilers provide an easy way to picture Meta-Classes. Classes must be implemented in some way; perhaps with dictionaries for methods, instances, and parents and methods to perform all the work of being a class. This can be declared in a class named "Meta-Class". The Meta-Class can also provide services to application programs, such as returning a set of all methods, instances or parents for review (or even modification). [Booch 91, p 119] provides another example in Smalltalk with timers. In Smalltalk, the situation is more complex

Question: What Is Inheritance?

Answer: Inheritance provides a natural classification for kinds of objects and allows for the commonality of objects to be explicitly taken advantage of in modeling and constructing object systems. Natural means we use concepts, classification, and generalization to understand and deal with the complexities of the real world. See the example below using computers. Inheritance is a relationship between classes where one class is the parent base/superclass/ancestor/etc.) class of another. Inheritance provides programming by extension (as opposed to programming by reinvention [LaLonde 90]) and can be used as an is-a-kind-of (or is-a) relationship or for differential programming. Inheritance can also double for assignment

Question: What Is The Difference Between Object-Based And Object-Oriented?

Answer: Object-Based Programming usually refers to objects without inheritance [Cardelli 85] and hence without polymorphism, as in '83 Ada and Modula-2. These languages support abstract data types (Adts) and not classes, which provide inheritance and polymorphism. Ada95 and Modula-3; however, support both inheritance and polymorphism and are object-oriented. [Cardelli 85, p481] state "that a language is object-oriented if and only if it satisfies the following requirements: - It supports objects that are data abstractions with an interface of named operations and a hidden local state. - Objects have an associated type. - Types may inherit attributes from supertypes. object-oriented = data abstractions + object types + type inheritance These definitions are also found in [Booch 91, Ch2 and Wegner 87]. [Coad 91] provides another model: Object-Oriented = Classes and Objects + Inheritance + Communication with messages.

Glossary (Updated Version)

Abstract class : A class that can only be used as a base class for some other class. A class is abstract if it has at least one pure virtual function.

Access control : A C++ mechanism for prohibiting or granting access to individual members of a class. See public, private, protected, and visibility.

Access declaration : A way of controlling access to a specified member of a base class when it is used in a derived class.

Access specifier : A way of labelling members of a class to specify what access is permitted i.e public, private, and protected.

Accessor : : A public member subprogram that provides query access to a private data member.

Agent: : An object that can both initiate behavior in other objects, as well as be operated upon by other objects.

Allocation: The process of giving memory space to an object. See dynamic storage,static storage, and deallocation.

ANSI: Acronym for American National Standards Institute, a standards body currently standardizing C++.

array : : An ordered collection that is indexed.

array constructor: A means of creating a part of an array by a single statement.

array overflow: : An attempt to access an array element with a subscript outside the array size bounds.

Array pointer: : A pointer whose target is an array, or an array section.

Array section: : A subobject that is an array and is not a defined type component.

Assertion: : A programming means to cope with errors and exceptions.

Assignment operator: : The equal symbol, “=”, which may be overloaded by a user.

Attribute: : A property of a variable that may be specified in a type declaration statement.
Base class: : A previously defined class whose public members can be inherited by another class. (Also called a super class.)
Behavior sharing: : A form of polymorphism, when multiple entities have the same generic interface. This is achieved by inheritance or operator overloading.
Binary operator: : An operator that takes two operands.
Binary tree: : A tree structure where each node has two child nodes.
Call-by-reference: : A language mechanism that supplies an argument to a procedure by passing the address of the argument rather than its value. If it is modified, the new value will also take effect outside of the procedure.
Call-by-value: : A language mechanism that supplies an argument to a procedure by passing a copy of its data value. If it is modified, the new value will not take effect outside of the procedure that modifies it.
Class attribute: : An attribute whose value is common to a class of objects rather than a value peculiar to each instance of the class.
Class descriptor: : An object representing a class, containing a list of its attributes and methods as well as the values of any class attributes.
Class diagram: : A diagram depicting classes, their internal structure and operations, and the fixed relationships between them.
Class inheritance: : Defining a new derived class in terms of one or more base classes.
Class: : An abstraction of an object that specifies the static and behavioral characteristics of it, including their public and private nature. A class is an ADT with a constructor template from which object instances are created.
Concrete class: : A class having no abstract operations and can be instantiated.
Constructor: : An operation, by a class member function, that initializes a newly created instance of a class.
Container class: : A class whose instances are container objects. Examples include sets, arrays, and stacks.
Container object: : An object that stores a collection of other objects and provides operations to access or iterate over them.
Data hiding: : The concept that some variables and/or operations in a module may not be accessible to a user of that module; a key element of data abstraction.
Information hiding: : The principle that the state and implementation of an object should be private to that object and only accessible via its public interface.
Inheritance: : The relationship between classes whereby one class inherits part or all of the public description of another base class, and instances inherit all the properties and methods of the classes which they contain.
Instance: : A individual example of a class invoked via a class constructor.
Linked list: : A data structure in which each element identifies its predecessor and/or successor by some form of pointer.
Member data: : Variables declared as components of a defined type and encapsulated in a class.
Member function: : Subprograms encapsulated as members of a class.
message passing: : The philosophy that objects only interact by sending messages to each other that request some operations to be performed.
Message: : A request, from another object, for an object to carry out one of its operations.

Method: : A class member function encapsulated with its class data members.
object: : A concept, or thing with crisp boundaries and meanings for the problem at hand; an instance of a class.
Object diagram: : A graphical representation of an object model showing relationships, attributes, and operations.
Object-oriented (OO): : A software development strategy that organizes software as a collection of objects that contain both data structure and behavior.
object-oriented programming (OOP) : Object-oriented programs are object-based, class-based, support inheritance between classes and base classes and allow objects to send and receive messages.
Operation: : Manipulation of an object's data by its member function when it receives a request.
Operator overloading: : A special case of polymorphism; attaching more than one meaning to the same operator symbol. 'Overloading' is also sometimes used to indicate using the same name for different objects.
Overloading: : Using the same name for multiple functions or operators in a single scope.
Overriding: : The ability to change the definition of an inherited method or attribute in a subclass.
Parameterized classes: : A template for creating real classes that may differ in well-defined ways as specified by parameters at the time of creation. The parameters are often data types or classes, but may include other attributes, such as the size of a collection. (Also called generic classes.)
Pointer: : A single data object which stands such as an array, or defined type.
Polymorphism: : The ability of an function/operator, with one name, to refer to arguments, or return types, of different classes at run time.
Private: : That part of an class, methods or attributes, which may not be accessed by other classes, only by instances of that class.
Protected: : (Referring to an attribute or operation of a class in C++) accessible by methods of any descendent of the current class.
Public: : That part of an object, methods or attributes, which may be accessed by other objects, and thus constitutes its interface.
Super class: : A class from which another class inherits.

==========

Spring 2011 Latest Papers (Current)

Papers Number 01

Write c++ code for operator() ……5 marks

20.3. Overloading Function () operator

Properties of Functions are:

- Must be a member function
- Any number of parameters can be specified
- Any return type can be specified
- Operator() can perform any generic operation

Function Operator

```
class String{
...
```

```
public:
char & operator()(int);
...
};
char & String::operator()
(int pos){
assert(pos>0 && pos<=size);
return bufferPtr[pos-1];
}
int main(){
String s1("Ping");
char g = s1(2); // g = 'i'
s1(2) = 'o';
cout << g << "\n";
cout << str.GetString();
return 0;
}
```

Output:

```
i
Pong
```

Write c++ code for operator[] using string class.....5 marks.....

20.2. Overloading Subscript [] Operator

Subscript operator must be overloaded as member function of the class with one parameter of integer type,

```
class String{
...
public:
char & operator[](int);
...
};
char & String::operator[]( int pos){
assert(pos>0 && pos<=size);
return stringPtr[pos-1];
}
int main() {
String s1("Ping");
cout <<str.GetString()<< endl;
s1[2] = 'o';
cout << str.GetString();
return 0;
}
```

Output:

```
Ping
```

Pong

Static data members.....2.marks

Static Data Member

Definition

"A variable that is part of a class, yet is not part of any object of that class, is called static data member". They are shared by all instances (objects) of the class.

They do not belong to any particular instance of a class

Class vs. Instance Variable

Suppose we created three objects of student class as shown below,

Student s1, s2, s3;

Static Data Member (Syntax)

Keyword static is used to make a data member static

```
class ClassName{
…
static DataType VariableName;
};
```

Defining Static Data Member

Static data member is declared inside the class

But they are defined outside the class

Defining Static Data Member

```
class ClassName{
…
static DataType VariableName;
};
DataType ClassName::VariableName;
```

Initializing Static Data Member

Static data members should be initialized once at file scope

They are initialized at the time of definition

Example

```
class Student{
private:
static int noOfStudents;
public:
…
};
int Student::noOfStudents = 0;
/*private static member cannot be accessed outside the class except for initialization*/
```

Initializing Static Data Member

If static data members are not explicitly initialized at the time of definition then they are initialized to 0

Example

```
int Student::noOfStudents;
```

is equivalent to
int Student::noOfStudents=0;

Papers Number 02

Question No: 1 (Marks: 1)
Inheritance is a way to
- ► organize data.
- ► pass arguments to objects of classes.
- ► **add features to existing classes without rewriting them. Pg 27**
- ► improve data-hiding and encapsulation.

Question No:2 (Marks: 1)
To convert from a user-defined class to a basic type, you would most likely use

- ► a built-in conversion operator.
- ► a one-argument constructor.
- ► **an overloaded = operator.**
- ► a conversion operator that's a member of the class.

Question No:3 (Marks: 1)
A C++ class is similar to ----------

- ► **Structure**
- ► Header File
- ► Library File
- ► None of the given

Question No:4 (Marks: 1)
a'A static member function can be called, even when a class is not _____.

- ► Declared
- ► Define
- ► **Instantiated**
- ► Called

Question No:5 (Marks: 1)
Which one of the following features of OOP is used to derive a class from another?

- ► Encapsulation
- ► Polymorphism
- ► Data hiding
- ► **Inheritance**

Question: 6 (Marsk:1)
What is a class?

►A class is a section of computer memory containing objects
►A class is a section of the hard disk reserved for object oriented programs
►A class is the part of an object that contains the variables.

►**A class is a description of a kind of object**

Question: 7 (Marks:1)
The main function of scope resolution operator (**::**) is,
►To define an object
►**To define a data member**
►To link the definition of an identifier to its declaration
►All of the given

Question No: 8 (Marks: 1)
Which of the following operators always takes no argument if overloaded?

► /
► -
► +
► **++**

Question No: 9 (Marks: 1)
Assume a class C with objects obj1, obj2, and obj3. For the statement obj3 = obj1 - obj2 to work correctly, if the overloaded - operator must

► **take two arguments.**
► return a value.
► create a named temporary object.
► take four arguments

Question: 10 (Marks:1)
The keyword that is used that the variable can not change state?
►static
►**const**
►friend
►private

Question No:11 (Marks: 1
Suppose obj1 and obj2 are two objects of a user defined class A. An + operator is overloaded to add obj1 and obj2 using the function call obj1+obj2.
Identify the correct function prototype against the given call?

► A operator + (A &obj);
► int + operator();
► **int operator (plus) ();**
► A operator(A &obj3);

Question No: 11(Marks: 3)

Explain what type of copy the default assignment operator "=" does when applied to objects. (shallow copy or deep copy)

09.1. Shallow Copy

When we initialize one object with another then the compiler copies state of one object to the other using copy constructor by assigning data member values of previous object to newly created object. This kind of copying is called shallow copying.

Shallow copy using default Copy Constructor (Syntax)

```
Student::Student( const Student & obj ){
rollNo = obj.rollNo;
name = obj.name;
GPA = obj.GPA;
}
```

This kind of copying is called shallow copying and is called default method for copying.

Write the code for Deep copy constructor for the given class.

In deep copy we write copy constructor code by our self to ensure that when one object is copied from other object new dynamic memory is allocated for it as well so that it doesn't rely on previous object memory.

Deep Copy:

Let we first write code for deep copy in our base class Person.For this we will write our own copy constructor as we wrote our own default constructor as shown below,

```
Person::Person(const Person & rhs): name(NULL){
/* Code for deep copy*/
if (rhs.name != NULL)
{
name = new char[strlen(rhs.name)+1];
strcpy(name,rhs.name);
}
}
int main(){
Student sobj1("Ali","Computer Science");
Student sobj2 = sobj1;
//Student sobj2(sobj1); Similar to above statement to call copy constructor
sobj2.Print();
system("PAUSE");
return 0;
}
```

Question No: 12(Marks: 5)

How we can overload Stream Extraction and Insertion Operators in c++?Give example code for Complex Number Class.(pg#156)

Overloading Stream Insertion and Stream Extration Operators:

```
class Complex{
...
friend ostream & operator << (ostream & os, const Complex & c);
};
```

Stream Insertion operator

```
// we want the output as: (real, img)
ostream & operator << (ostream & os, const Complex & c){
os << '(' << c.real
<< ','
```

<< c.img << ‘)’;
return os;
}
ostream reference can not be const as it store the data in its buffer to insert on output stream, however Complex reference will be constant as we are only getting data from Complex object and inserting it to output stream.
Complex c1(1.01, 20.1), c2(0.01, 12.0);
cout << c1 << endl << c2;
Stream Insertion operator
Output:
(1.01 , 20.1)
(0.01 , 12.0)
Now cascading statements are also possible as given below,
cout << c1 << c2;
is equivalent to operator<<(**operator<<(cout,c1)**,c2);
Same thing can be done with stream extraction operator,
Stream Extraction Operator Code:
istream & operator << (istream & in, Complex & c){
in >> c.real;
in >> c.img;
return in;
}
Main Program:
Complex c1(1.01, 20.1);
cin >> c1;
// suppose we entered // 1.0025 for c1.real and // 0.0241 for c1.img
cout << c1;
Output:
(1.0025 , 0.0241)

Question:13(Marks:3)

Justify your answer with example could it possible “Array of Objects”?

12.6. Array of Objects
• Array of objects can only be created if an object can be created without supplying an explicit initializer.
• There must always be a default constructor if we want to create array of objects
12.4. this Pointer and static member functions
• this pointer is passed implicitly to member functions
• this pointer is not passed to static member functions
• Reason is static member functions cannot access non static data members
12.5. Global Variable vs. Static Members
• Alternative to static member is to use global variable
• Global variables are accessible to all entities of the program
• User of Global variables is against the principle of information hiding.

Question:14 (Marks:4)

Explain with example post-fix and pre-fix operator?()

Unary operators are usually prefix, except for **++** and **--**

++ and -- both act as prefix and postfix
Example:
h++;
g-- + ++h - --i;
General syntax for unary operators
As Member Functions:
TYPE & operator OP (); // no argument the object with respect to which it is called is taken as one operand
As Non-member Functions:
Friend TYPE & operator OP (TYPE & t);
// one argument object with respect to which it is called.
Object Oriented Programming (CS304) VU
© Virtual University of Pakistan 163
Overloading unary '-':

```
class Complex{
...
Complex operator - ();
// friend Complex operator
// -(Complex &);
}
Complex Complex::operator -(){
Complex temp;
temp.real = -real;
temp.img = -img;
return temp;
}
```

Complex c1(1.0 , 2.0), c2;
c2 = -c1;
// c2.real = -1.0
// c2.img = -2.0
Unary '+' is overloaded in the same way.
A post-fix unary operator is implemented using:
Member function with 1 dummy int argument
OR
Non-member function with two arguments
In post increment, current value of the object is stored in a temporary variable
Current object is incremented
Value of the temporary variable is returned

Papers Number 03

Explain the statement below, ?
vector<int> ivec(4, 3);

A **Vector** class template can store data elements of different types, without templates, we need a separate **Vector** class for each data type.
We can create this vector class instances for int or char data type as given below,
Vector< int > intVector;

Vector< char > charVector;
This Vector class is parameterized class and will always be instantiated for a particular type only. Now we can not create object of type Vector only it will be instantiated for a particular data type like Vector<int> or Vector <float> and so on…

Give the names of any two types of template?

In C++ generic programming is done using templates.
Templates are of two kinds,
a. **Function Templates** (in case we want to write general function like printArray)
b. **Class Templates** (in case we want to write general class like Array class)
Compiler generates different type-specific copies from a single template
This concept is similar to concept of making prototype in the form of class for all objects of same kind.

Describe the way to declare a template function as a friend of any class?

When we declare a template as friend of any class then all kinds' specializations of that function – explicit, implicit and partial, also becomes friends of the class granting friendship.

What are binary operators? Give an example of binary operators overloading using any class?

Binary Operators Overloading:
Binary operators act on two quantities.
Examples of binary operators:

+	-	*	/	%	^	&	|	~
!	=	<	>	+=	-=	*=	/=	%=
^=	&=	|=	<<	>>	>>=	<<=	==	!=
<=	>=	&&	||	,	->*	->		

Examples:
Overloading + operator:

```
class Complex{
private:
double real, img;
public:
…
Complex operator +(const Complex & rhs);
};
Complex Complex::operator +( const Complex & rhs){
Complex t;
t.real = real + rhs.real;
t.img = img + rhs.img;
return t;
}
```

The binary operator is always called with reference to the left hand argument.

Detect and correct compile time error(s) in the following code?

```
class  Exam
{
     char *ExamName;    //string should be used
     int No_of_paper;

public:
      Exam()
      {
           ExamName = "Final Term";
           No_of_paper = 5;
           }

void setname( char* name)  const              //delete const
{
    ExamName = name;
    }
void setpaper(int paper)   const        //delete const
{
    No_of_paper = paper;
    }
Const char* getname()
{
     return ExamName;
     }
Const int getpaper()
{
   return No_of_paper;
}
 };

int main()
{
const Exam exam1;                        //Delete Const keyword

cout << " Exam           = "<<exam1.getname()<<endl;
cout << " Numbe of paper  = " << exam1.getpaper();

getch();
return 0;
}
```

Answer (Corrected Working Version)

```
class  Exam
{
     char *ExamName;    //string should be used
     int No_of_paper;

public:
```

```
        Exam()
        {
              ExamName = "Final Term";
              No_of_paper = 5;
              }

void setname( char* name)              //delete const
{
     ExamName = name;
     }
void setpaper(int paper)  //delete const
{
     No_of_paper = paper;
     }
const char* getname()
{
      return ExamName;
      }
const int getpaper()
{
    return No_of_paper;
}
 };

int main()
{
Exam exam1;

cout << " Exam            = "<<exam1.getname()<<endl;
cout << " Numbe of paper  = " << exam1.getpaper();

getch();
return 0;
}
```

Papers Number 04

Solved Mid-Term Past Papers

Short Question (Set-1)

Question No. 2 Marks : 05

a. Write the exact type of ***this*** pointer in a ***member function*** of a class **XYZ**. **02**

b. Write three distinct situations in which copy constructor of a class is called. **03**

Answer:

a)

XYZ *this;

b)

1. Assignment of private data members at the time of object creation.

2. When an object is passed by vale to a function.

3. When allocating memory dynamically we use copy constructor to avoid dangling pointer issue.

Question No. 3 Marks : 05

```
class Complex
{
private:
double x,y;
static int z;
public:
Complex(double = 0.0);
friend ostream& operator<<(ostream&, const Complex&);
static int doSomething( ) { z = 2 * y; return z; }
};
```

a. What is wrong in the definition of member function **doSomething( ). 03**

Answer:

Static keyword because using static key word we are declaring a static member function doSomething(). Which can not use non static data members.

b. What will be the effect of writing the friend function **operator<<(...)** in **private** part of the above class? **02**

Answer:

There will be no effect on friend function if we write it in private part. Friend function is a friend function and can use any private or public data member of the class. Where ever we declare it in class body.

Question No: 6 (Marks: 5)

Write any two advantage(s) of declaring a member function as const?

Constant member functions

- This will enable constant objects to access more member functions
- Allow to access const data member of the class.
- It does not allow function to change the value of data member.

==========>

Short Question (Set-2)Q.

What is difference between simple association and composition?

Object Association

It is the interaction of stand alone objects of one class with other objects of anther class. It can be of one of the following types,

• Simple Association
• Composition
• Aggregation

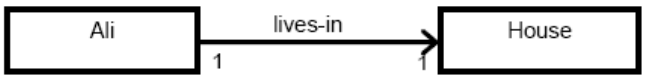
05.3. Simple Association

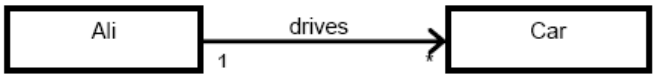
The two interacting objects have no intrinsic relationship with other object. It is the weakest link between objects. It is a reference by which one object can interact with some other object.

Ali lives in a house ,Ali drives a car

It is generally called as "association" instead of "simple association"

Ali lives-in House





Kinds of Simple Association

Simple association can be categorized in two ways,

- With respect to direction (navigation)
- With respect to number of objects (cardinality)

Kinds of Simple Association w.r.t Navigation

With respect to navigation association has the following types,

a. One-way Association

b. Two-way Association

a. One-way Association

In One way association we can navigate along a single direction only, it is denoted by an arrow towards the server object.

Examples:

- Ali lives in a House
- Ali drives his Car

b. Two-way Association

In two way association we can navigate in both directions, it is denoted by a line between the associated objects

Examples:

Employee works for company

Company employs employees

Two-way Association - Example

Yasir is a friend of Ali

Ali is a friend of Yasir

Kinds of Simple Association w.r.t Cardinality

With respect to cardinality association has the following types,

a. Binary Association

b. Ternary Association

c. N-ary Association

a. Binary Association

It associates objects of exactly two classes; it is denoted by a line, or an arrow between the associated objects.

Example

Association "works-for" associates objects of exactly two classes

Association "drives" associates objects of exactly two classes

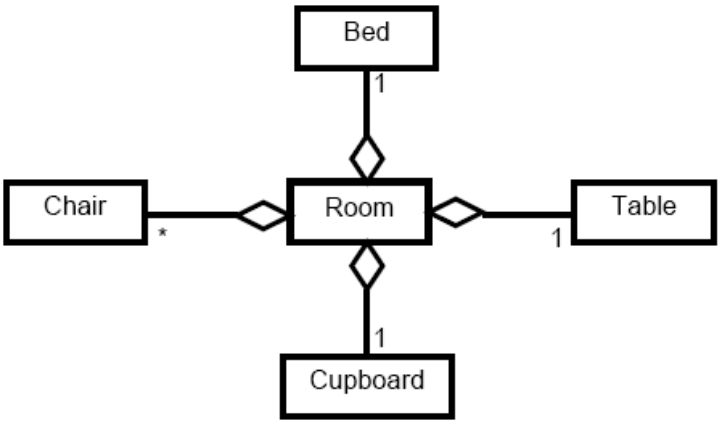
b. Ternary Association
It associates objects of exactly three classes; it is denoted by a diamond with lines connected to associated objects.
Example
Objects of exactly three classes are associated
c. N-ary Association
An association between 3 or more classes its practical examples are very rare.

05.4. Composition

An object may be composed of other smaller objects, the relationship between the "part" objects and the "whole" object is known as Composition, Composition is represented by a line with a filled-diamond head towards the composer object.
Example – Composition of Ali by its different body parts.
Composition is a stronger relationship, because
Composed object becomes a part of the composer
Composed object can't exist independently
Example I
Ali is made up of different body parts. They can't exist independent of Ali
Example II
Chair's body is made up of different parts. They can't exist independently

05.5. Aggregation

An object may contain a collection (aggregate) of other objects, the relationship between the container and the contained object is called aggregation. Aggregation is represented by a line with unfilled-diamond head towards the container.
Example – Aggregation



Aggregation is a weak relationship, because
Aggregation object does not becomes a part of the container
Aggregation object can exist independently

Q.

Friend functions minimize "Encapsulation", What is your opinion?

15.2. Friend Functions
The functions which are not member functions of the class yet they can access all private members of the class are called friend functions.
It can be said that friend functions are against the principle of object oriented programming because they violate the principle of encapsulation and information hiding which clearly says that each object methods and functions should be encapsulated in it. But there we are making our private member accessible to other outside functions.

Q.

Write three important properties of constructors?

08.5. Constructor Properties

The properties of Constructors are given below.

- Constructor is a special function having same name as the class name
- Constructor does not have return type
- Constructors are commonly public members

1. Tell the reason why we can not overload the following four operators in c++
. , .* , :: , ?:

Reason: They take actual current object name, rather than value in their argument as you may have seen in the use of dot ('.') operator,

2. Considering the complex number class can we subtract two complex numbers by overloading plus "+ "Operator. Justify your answer as well.

Yes, we can do it and compiler will make no error. But it will be difficult for the user to manipulate. He will get wrong error according to the method .

Q.

How we resolve the following problems in overloading of assignment operator in string class, (explain with the help of c++ code)

a. Self referencing

18.1. Self assignment problem:

In we assign same string to itself as done in main function below our program will produce unexpected results as source and destination operands for copying are same,

```
int main(){
String str1("Fakhir");
str1 = str1; // Self Assignment problem…
return 0;
}
```

Result of str1 = str1

b. Assigning a string value to more than one strings in a single line like, stringobject1 = string object2 = stringobject3 = stringobject4

Q.

Consider the class given below explain the order in which variables x,y and z will be initialized after creating object of this class,

```
class XYZ{
        int e;
        int f;
        int g;
public:
```

```
        XYZ ();
};

XYZ:: XYZ ():g(30),e(10),f(20)
{
        …
}
```

Data member of any class are initiated in the order they are defined in the class.

==========>

Short Question (Set-3)

Question No: 19 (Marks: 2)

Explain two benefits of setter functions.

Objects can make their data invisible (in accordance with the principle of data hiding). Setters and getters functions are provided by class to access the its members it also minimizes the changes to move the objects in inconsistent state as we can write checks in our setter functions for example

- We can check that whether the user has entered correct age value and has not entered negative value for age.
- Object remains in consistent state.

It is against the principle of OOP to access the data members directly using object of class as we have done above. This code is given for example only we should write assessor functions (setters and getters) wherever we want to access the members of the class.

Question No: 22 (Marks: 5)

What is composition? Explain it with the help of an example.

Answered Above

Question No: 23 (Marks: 5)

How we can overload Stream Extraction and Insertion Operators in c++?Give example code for Complex Number Class.

Answered Above.

==========>

Short Question (Set-7)

Question No: 31 (Marks: 1)

Write the syntax of declaring a pure virtual function in a class?

Ans:
Pure Virtual Function is a Virtual function with no body.

Declaration of Pure Virtual Function:
Since pure virtual function has no body, the programmer must add the notation =0 for declaration of the pure virtual function in the base class.

General Syntax of Pure Virtual Function takes the form:

```
class classname //This denotes the base class of C++ virtual function
{
public:
virtual void virtualfunctioname() = 0 //This denotes the pure virtual function in C++
};
```

Question No: 32 (Marks: 1)

What is meant by direct base class ?

Ans

When a class-type is included in the class-base, it specifies the direct base class of the class being declared. If a class declaration has no class-base, or if the class-base lists only interface types, the direct base class is assumed to be object. A class inherits members from its direct base class,

Deriving a class from more than one *direct base class* is called multiple inheritance.

Question No: 33 (Marks: 2)

Describe the way to declare a template class as a friend class of any other class.

Ans

The following example is use of a class template:

```
template<class L> class Key
{
    L k;
    L* kptr;
    int length;
public:
    Key(L);
    // ...
};
```

Suppose the following declarations appear later:

```
Key<int> i;
  Key<char*> c;
  Key<mytype> m;
```

The compiler would create three objects.

Question No: 34 (Marks: 2)

What is the purpose of template parameter?

Ans: There are three kinds of template parameters:

- type
- non-type
- template

You can interchange the keywords **class** and **typename** in a template parameter declaration. You cannot use storage class specifiers (**static** and **auto**) in a template parameter declaration.

Question No: 35 (Marks: 3)

Describe in simple words how we can use template specialization to enforce case sensitive specialization in String class.

Ans:
The act of creating a new definition of a function, class, or member of a class from a template declaration and one or more template arguments is called template instantiation. The definition created from a template instantiation is called a specialization. A primary template is the template that is being specialized.
create function objects to do the case-insensitive compares, and
then reuse them when also wanting to do case-insensitive sorting
or searching.

Question No: 36 (Marks: 3)

Can we use compiler generated default assignment operator in case our class is using dynamic memory? Justify your answer.

Ans: The compiler does not make a separate copy of the object. Even if the types are not the same, the compiler is usually able to do a better job with initialization lists than with assignments.
Consider the following constructor that initializes member object x_ using an initialization list: square::square() : x_(*whatever*) {}. The most common benefit of doing this is improved performance. For example, if the expression *whatever* is the same type as member variable x_, the result of the *whatever* expression is constructed directly inside x_ — the compiler does not make a separate copy of the object. Even if the types are not the same, the compiler is usually able to do a better job with initialization lists than with assignments.
As if that wasn't bad enough, there's another source of inefficiency when using assignment in a constructor: the member object will get fully constructed by its default constructor, and this might, for example, allocate some default amount of memory or open some default file. All this work could be for naught if the *whatever* expression and/or assignment operator causes the object to close that file and/or release that memory (e.g., if the default constructor didn't allocate a large enough pool of memory or if it opened the wrong file).

Question No: 37 (Marks: 3)

Give the names of three ways to handle errors in a program.

Ans : The function will throw Divide ByZero as an exception that can then be caught by an exception-handling catch statement that catches exceptions of type int. The necessary construction for catching exceptions is a try catch system. If you wish to have your program check for exceptions, you must enclose the code that may have exceptions thrown in a try block.

The catch statement catches exceptions that are of the proper type. You can, for example, throw objects of a class to differentiate between several different exceptions. As well, once a catch statement is executed, the program continues to run from the end of the catch.
the errors can be handled outside of the regular code. This means that it is easier to structure the program code, and it makes dealing with errors more centralized. Finally,

because the exception is passed back up the stack of calling functions, you can handle errors at any place you choose.

Question No: 38 (Marks: 5)

Consider the following code,

```
class Base{
private:
void base1();
protected:
void base2();
public:
void base3();
};

class Derived: public Base{
private:
void derived1();
protected:
void derived2();
public:
void derived3();
};

int main(){
Derived * derived = new Derived();
return 0;
}
```

Fill the table below to tell which member functions of Base and Derived classes we can access using the **Derived** pointer in the code indicated in bold.

Ans:

Function Name	Availability (Yes / No)?
base2()	no
base3()	yes
derived1()	No
derived2()	No
derived3()	Yes

Question No: 39 (Marks: 5)

What is the output produced by the following program?

```
#include<iostream.h>
void sample_function(double test) throw (int);
int main()
{
        try
        {
                cout <<"Trying.\n";
                sample_function(98.6);
                cout << "Trying after call.\n";
        }
        catch(int)
        {
                cout << "Catching.\n";
        }

        cout << "End program.\n";
        return 0;
}
void sample_function(double test) throw (int)
{
        cout << "Starting sample_function.\n";
        if(test < 100)
           throw 42;
}
```

Ans:
Starting sample_function

Trying

Trying after call

Catching

End program

Question No: 40 (Marks: 10)
Write a publicly derived class "E**mployee"** that is derived from base class named "**Company".** Both classes will have function **"create()"**. Make virtual function of base class and override same function in derived class. Function create will have an output statement of your own choice.
In "**main**" Create an object of base class and call both functions with same object type.

Question No: 41 (Marks: 10)
Write a program in C++ which creates three classes named as

1. **Equation**
2. **Linear**
3. **Quadratic**

Where Linear and Quadratic are inherited from Equation
Each class has the method Graph. Graph method should be pure virtual in Equation class.
This method should be overridden in both the inherited classes. It is meant to display the Graph shape of its respective class. Graph method of Linear will display the message;
Straight line
Similarly, the Graph method of Quadratic will display the message;
Parabola
In main, call the Graph method of both the Linear and Quadratic equations polymorphically through the parent class (Equation).
Ans:

```
#include "fraction.h"
#include <iostream>
#include <string>
#include <string.h>
#include <stdlib.h>
class equation;

class equation {
    int a, b;
  public:
    int c ()
      {return (c);}
    void convert (Cequation);
};

class linear {
  private:
    int side;
  public:
    void set_side (int a)
      {side=a;}
    friend class equation;
};

void equation::convert (Cequation) {
  a = 23;
  b = 45;
}

int main () {
  cequation sqr;
  CRectangle rect;
  sqr.set_side(4);
  rect.convert(sqr);
  cout << rect.area();
```

```
  return 0;
}
```

==========>

Short Question (Set-8)

Question No: 27 (Marks: 2)

Describe the way to declare a template function as a friend of any class.

Template templatename
Class calssname
{
Friend void friend templatename (classname <templatename> astric const prt classname);
}

Question No: 28 (Marks: 2)

State any two reasons why the virtual methods can not be static?

1-virtual method can not be static as it is dynamic
2-as virtual method is dynamic so it works automatically that is also another reason That virtual method can not be static.

Question No: 29 (Marks: 2)

Explain the statement below,

vector<int> ivec(4, 3);

Question No: 30 (Marks: 2)

Explain two benefits of setter functions.

1- It minimize the changes to move the objects in inconsistent states
2- You can write checks in your setter functions to check the validity of data entered by the user, for example age functions to check to calculate the age from date entered.

Question No: 31 (Marks: 3)

Consider the code below,

```
template< typename T >
class T1 {
    public:
    T i;
    protected:
    T j;
    private:
    T k;
    friend void Test();
    };
```

This code has a template class T1 with three members i,j and k and a friend function Test(), you have to describe which member/s of T1 will be available in function Test().

```
public:
    T i;
    protected:
    T j;
```

Question No: 32 (Marks: 3)

What do you mean by Stack unwinding?

When we want to check what happens actually to the local variables in the try block when then an exception is thrown this concept is called stack unwinding.

Question No: 33 (Marks: 3)

Give the c++ code of case sensitive comparison function of string class.

Question No: 34 (Marks: 5)

What is random_iterator? What is relation between random_iterator and Vector?

Random_iterator: it provided both increment and decrement and also provide constant time methods for moving forward and backword in arbitrary sized steps. Ramdom iterator provide asentially all of the operations of ordinary c pointer arithmetic.

Vector class provide an stl style random access iterator for use with generic algorithm since neither the vactor nor the matrix classes are container classes in actuall. The iterator class is really an iterator of data object that is viewed by vector or matrix.

Question No: 35 (Marks: 5)

What would be the output of this code?

```
class mother {
  public:
    mother ()
```

```
    { cout << "mother: no parameters\n"; }
  mother (int a)
    { cout << "mother: int parameter\n"; }
};

class daughter : public mother {
  public:
    daughter (int a)
    { cout << "daughter: int parameter\n\n"; }
};

class son : public mother {
  public:
    son (int a) : mother (a)
    { cout << "son: int parameter\n\n"; }
};

int main () {
  daughter rabia (0);
  son salman(0);

  return 0;
}
```

Output will be

```
mother: no parameters
daughter: int parameter

mother: int parameter
son: int parameter
```

Question No: 36 (Marks: 5)

The code given below has one template function as a friend of a template class,

1. You have to identify any error/s in this code and describe the reason for error/s.
2. Give the correct code after removing the error/s.

```
template<typename U>
void Test(U);
template< class T >

class B {
        int data;
        public:
        friend void Test<>( T );
};

template<typename U>
```

```
void Test(U u){
    B < int> b1;
    b1.data = 7;
    }
int main(int argc, char *argv[])
{
    char i;
    Test(i);
    system("PAUSE");
    return 0;
}
```

==========>

Short Question (Set-10)

q.1 can constant object access the none constant member function of the class.

No, they can't access

q.2. Give at least two problems that we should check when we overloading assignemts oprater ("=") in string class

Q3. Give c++ code to overloaded unary "--" oprators to comples member class.

q4.What is simple association? explain it with the help of example.

Answered Above.....

Q5.explain the defferent between the static variable of a class with none static variable with the help of example

==========>

Short Question (Set-11)

Power "^" operator overloading for complex class ---5

Use of Static member function with example-------5

Operator overloading for string class ----------3

A code was given for "class A" and "class B" about which we must tell the association relation between classes ---------3

Why Static member functions are used to retrieve static data members.-------2

==========>

Short Question (Set-12)

Give at least two problems that we should check when we overloading assignment operator ("=") in string class.

Suppose we have a parent class and a child class , give the order in which constructor and destructor of there classes.

Briefly describe the use of static data member with the help of example.

==========>

Short Question (Set-14)

==========>

Solved_MCQz

MCQz (Set-1)

We can get only one unique value which can be used by all the objects of that class by the use of,

static variables
dynamic variables
instance variables
data members

A member function having the same name as that of a class and a ~ sign with it is called,

Constructor
Getter
Setter
Destructor

Using encapsulation we can achieve
Information hiding
Least interdependencies among modules
Implementation independence
All of given options

Inheritance is a way to

make general classes into more specific classes.
pass arguments to objects of classes.
improve data hiding and encapsulation.
providing class growth through natural selection.

Static variable can be initialized more than once.
True
False

For classes with common behavior, you can save effort by placing the common behavior in a _____.
Derived Class
Base class
Deprived Class
Named class

Which of the following are an advantage of OOP?

OOP makes it easy to re-use the code
It provides an ability to create one user defined data type by extending the other
It provides the facility of defining Abstract data types through which real world entities can be defined better
All of the given options

The >= operator can be overloaded.

True
False

A static member function cannot be declared.
Static
Implicit
Explicit
Virtual

Static variables act like a global variable in the context or scope of the class.

True
False

The compiler won't object if you overload the * operator to perform division.

True
False

We can use "this" pointer in the constructor in the body and even in the initialization list of any class if we are careful,

TRUE
False

A C++ class is similar to ----------

Structure
Header File
Library File
None of the given

Which operator can not be overloaded?
The relation operator (>=)
Assignment operator (=)
Script operator ([])
Conditional operator (? :),Ternary Operator

An overloaded operator always requires one less argument than its number of operands.
True
False

A generalization-specialization relation between classes are implemented using

data hiding
friend classes
encapsulation
inheritance

In OOP a class is an example of _____
Data Type
Abstract Type
User Defined Type

None of the given

Identify which of the following overloaded operator function's declaration is appropriate for the given call?
Rational_number_1 + 2.325
Where Rational_number_1 is an object of user defined class Rational_number.
Rational_number operator+(Rational_number & obj);
Rational_number operator+(double& obj);
Rational_number operator+(Rational_number &obj, double& num);
operator+(double& obj);

A class can be identified from a statement by ----------
Noun
Pronoun
Verb
Adverb

The members of a class that can be accessed without creating the object of the class is called

Private member
Data Member
Public Member
Static

==========>

MCQz (Set-2)

Question No: 1 (Marks: 1) - Please choose one
Suppose there is an object of type Person, which of the following can be considered as one of its attributes
► Name
► Age
► Work()
► Both Name and Age

Question No: 2 (Marks: 1) - Please choose one
What a derived class can add?

► New data members
► New member functions and New friend functions
► New constructors and destructor
► All of given

Question No: 3 (Marks: 1) - Please choose one
_____ is/are used to access information hidden within an object?

► Interface
► Private data members
► Private member functions
► Both public and private members

Question No: 4 (Marks: 1) - Please choose one
this pointers are not accessible for static member functions.

► True
► False

Question No: 5 (Marks: 1) - Please choose one
A static member function cannot be declared.

► Static
► Implicit
► Explicit
► Virtual

Question No: 6 (Marks: 1) - Please choose one
C++ compiler does not allow to dynamically allocate memory for objects

► False
► True

Question No: 7 (Marks: 1) - Please choose one
Given the following class
class Base{
int Age=33;
}
How you can improve above class with respect to accessing the field Age?

► Define the variable Age as private
► Define the variable Age as protected
► Define the variable Age as private and create a get method that returns it and a set method that updates it
► Define the variable Age as protected and create a set method that returns it and a get method that updates it

Question No: 8 (Marks: 1) - Please choose one
Friend class and friend function can be used as an alternate to each other

► True
► False

Question No: 9 (Marks: 1) - Please choose one

Which of the following operators always takes no argument if overloaded?

► /
► -
► +
► ++

Question No: 10 (Marks: 1) - Please choose one
Suppose that the Test class does not have an overloaded assignment operator. What happens when an assignment a=b; is given for two Test objects a and b?

► The automatic assignment operator is used
► The copy constructor is used
► Compiler error
► Run-time error

Question No: 11 (Marks: 1) - Please choose one
Assume a class C with objects obj1, obj2, and obj3. For the statement obj3 = obj1 - obj2 to work correctly, if the overloaded - operator must

► take two arguments.
► return a value.
► create a named temporary object.
► take four arguments

Question No: 12 (Marks: 1) - Please choose one
Which operator can not be overloaded?

► The relation operator (>=)
► Assignment operator (=)
► Script operator ([])
► Conditional operator (? :)

Question No: 13 (Marks: 1) - Please choose one
We achieve independence of internal implementation from its external interface through----------

► Encapsulation
► Information Hiding
► Abstraction
► both encapsulation and information hiding

Question No: 14 (Marks: 1) - Please choose one
Which one of the following is not an object association?

► Simple Assocation
► Inheritance
► Aggregation
► Composition

Question No: 15 (Marks: 1) - Please choose one
We capture the object attributes and behavior in Object Oriented programming using----------

► **Class**
► Function
► Data Members
► Instances

Question No: 16 (Marks: 1) - Please choose one
The return type of a constructor is of ----------

► Integer
► Chracter
► Double
► **No type pg 74**

==========>

MCQz (Set-3)

Question No: 1 (Marks: 1) - Please choose one
Which part of an object exhibits its state?
► **Data**

► Operations

► Any public part

► Any private part

Question No: 2 (Marks: 1) - Please choose one
Inheritance is a way to
► organize data.
► pass arguments to objects of classes.
► **add features to existing classes without rewriting them.**
► improve data-hiding and encapsulation.

Question No: 3 (Marks: 1) - Please choose one
Suppose you have been given the following design,

"A person has a name, age, address and -----. You are designing a class to represent a type of person called a patient. This kind of person may be given a diagnosis, have a spouse and may be alive".
Given that the person class has already been created, what of the following would be appropriate to include when you design the patient class?

► **registration date and diagnosis**
► age and -----

► ----- and diagnosis
► diagnosis and age

Question No: 4 (Marks: 1) - Please choose one
What problem(s) may occur when we copy objects without using deep copy constructor?

► Dangling pointer
► Memory Leakage(Object is not deleted)
► All of the given
► System crash

Question No: 5 (Marks: 1) - Please choose one
this pointers are not accessible for static member functions.

► True
► False

Question No: 7 (Marks: 1) - Please choose one
_____ remain in memory even when all objects of a class have been destroyed.

► Static variables
► Instance variable
► Primitive variables
► None of given

Question No: 8 (Marks: 1) - Please choose one
Friend functions are _____ functions of a class.

► None of given
► object member
► non-member
► data member

Question No: 9 (Marks: 1) - Please choose one
_____, which means if A declares B as its friend it does NOT mean that A can access private data of B. It only means that B can access all data of A.

► Friendship is one way only
► Friendship is two way only
► NO Friendship between classes
► Any kind of friendship

Question No: 10 (Marks: 1) - Please choose one
The statement objA=objB; will cause a compiler error if the objects are of different classes.

► True

► False

Question No: 12 (Marks: 1) - Please choose one
Which operator can not be overloaded?
► The relation operator (>=)
► Assignment operator (=)
► Script operator ([])
► Conditional operator (? :)

Question No: 13 (Marks: 1) - Please choose one
To convert from a user-defined class to a basic type, you would most likely use
► a built-in conversion operator.
► a one-argument constructor.
► an overloaded = operator.
► a conversion operator that's a member of the class.

Question No: 14 (Marks: 1) - Please choose one
The technique in which we visualize our programming problems according to real life's problems is called
---
► structured programming
► object oriented Programming
► procedural programming
► non of the given

Question No: 15 (Marks: 1) - Please choose one
In object orientated programming, a class of objects cans _____ properties from another class of objects

► Utilize
► Borrow
► Inherit
► Adopt

Question No: 16 (Marks: 1) - Please choose one
A C++ class is similar to ----------

► Structure
► Header File
► Library File
► None of the given
==========>

MCQz (Set-4)

Information hiding can be achieved through_____.

1. Encapsulation, Inheritance
2. Encapsulation, Polymorphism
3. **Encapsulation, Abstraction**
4. Overloading

A good model is ..... related to a real life problem.
Select correct option:

1. Loosely
2. Openly

Closely

Which of the following features of OOP is used to derive a class from another?
Select correct option:

1. Encapsulation
2. Polymorphism
3. Data hiding

Inheritance

Which of the following is a weak relationship between two objects?
Select correct option:

1. Inheritance
2. Composition

Aggregation(the object can exist without master class like room and chair)

3. None of given

Data items in a class must be private.
Select correct option:

1. True

False

Which one is not a class association?
Select correct option:

1. Simple Association

Inheritance

2. Composition
3. Aggregation

Suppose there is an object of type Person, which of the following can be considered as one of its attributes
Select correct option:

1. Name
2. Age
3. Work()

Both Name and Age

Using encapsulation we can achieve
Select correct option:

Information hiding

1. Least interdependencies among modules
2. Implementation independence
3. All of given options

In constant member function the type of this pointer is:
Select correct option:

1. Constant pointer

Constant pointer to object

2. Constant pointer to class
3. Constant pointer to constant object

Which of the following is the way to extract common behavior and attributes from the given classes and make a separate class of those common behaviors and attributes?

Generalization

1. Sub-typing
2. Specialization
3. Extension

The ability to derive a class from more than one class is called

1. Single inheritance
2. Encapsulation

Multiple inheritance

3. Polymorphism

If MyClass has a destructor what is the destructor named?

1. MyClass

~MyClass

2. My~Class
3. MyClass~

Class abc{ ----- }; Is a valid class declaration?

yes

1. no

Without using Deep copy constructor, A _____ problem can occur

1. System crash
2. Memory Leakage
3. Dangling pointer

All of the given

If only one behaviour of a derived class is incompatible with base class, then it is:

1. Generalization

Specialization

2. Extension
3. Inheritance

Which of the following may not be an integral part of an object?

1. state
2. behavior

Protected data members

3. All of given

Only tangible things can be chosen as an object.

1. True

False

==========>

MCQz (Set-5)

Question # 1 of 10 (Start time: 09:26:23 PM) Total Marks: 1
Consider the code below, class class1{ public: void func1(); }; class class2 : protected class1 {}; Function func1 of class1 is _____ in class2,
Select correct option:

public

protected

private

none of the given options

Question # 2 of 10 (Start time: 09:27:52 PM) Total Marks: 1
In case of dynamic memory allocation in a class we should use,
Select correct option:

User defined default constructor

User defined copy constructor

Both of these

None of these

Question # 3 of 10 (Start time: 09:29:20 PM) Total Marks: 1
Consider the code below, class class1{ protected: int i; }; class class2 : public class1 {};
Then int member i of class1 is _____ in class2,
Select correct option:

public

protected

private

none of the given options

Question # 4 of 10 (Start time: 09:30:13 PM) Total Marks: 1
In private inheritance derived class pointer can be assigned to base class pointer in,
Select correct option:

Main function

In derived class member and friend functions

In base class member and friend functions

None of the given options

Question # 5 of 10 (Start time: 09:31:42 PM) Total Marks: 1
Child class can call constructor of its,
Select correct option:

Direct base class pg 209

Indirect base class

Both direct and indirect base classes

None of these

Question # 6 of 10 (Start time: 09:31:59 PM) Total Marks: 1
Consider the following two lines of code written for a class Student, 1. Student sobj1,sobj2; 2. sobj2 = sobj1; In line No.2 what constructor of Student class will be called,
Select correct option:

Default constructor of Student class.

Copy constructor of student class

Both default and copy constructer of Student class

No constructor will be called.

Question # 7 of 10 (Start time: 09:33:17 PM) Total Marks: 1
Consider the code below, class class1{ public: int i; }; class class2 : public class1 {}; Then int member i of class1 is _____ in class2,
Select correct option:

public

protected

private

none of the given options

Question # 8 of 10 (Start time: 09:33:54 PM) Total Marks: 1
We can call base class constructor from derived class constructor,
Select correct option:

From derived class constructor body

From the initializer list of derived class constructor

From any member function of derived class

We can not call the base class constructor

Question # 9 of 10 (Start time: 09:35:20 PM) Total Marks: 1
Adding a derived class to a base class requires fundamental changes to the base class.
Select correct option:

True

False

Question # 10 of 10 (Start time: 09:36:44 PM) Total Marks: 1
Public Inheritance represents,
Select correct option:

"IS A" relationship

"Has A" relationship

"IS Special Kind of" relationship

None of these options

==========>

MCQz (Set-6)

Quiz Start Time: 10:18 AM

Question # 1 of 10 (Start time: 10:18:05 AM) Total Marks: 1
Virtual functions allow you to
Select correct option:

create an array of type pointer-to-base class that can hold pointers to derived classes.

create functions that can never be accessed.

group objects of different classes so they can all be accessed by the same function code.

use the same function call to execute member functions of objects from different classes.

Question # 2 of 10 (Start time: 10:19:28 AM) Total Marks: 1
Which one of the following operators is used to access members of value data types,
Select correct option:
->
.
&
None of the given options

Question # 3 of 10 (Start time: 10:20:54 AM) Total Marks: 1
Consider the code below, class c1{}; class c2 : public c1 {}; class c3 : public c2 {}; Then c2 is,

Select correct option:

Direct base class of c3

Direct child class of c3

Direct base class of c1

None of these

Question # 4 of 10 (Start time: 10:22:09 AM) Total Marks: 1
Which one of the following operators is used to access members of reference data types,
Select correct option:

->

.

&

None of the given options

Question # 5 of 10 (Start time: 10:23:19 AM) Total Marks: 1
In specialization we can,
Select correct option:

Replace child class with its base class

Replace base class with its child class

Replace both child and base classes interchangeably

None of the given options

Question # 6 of 10 (Start time: 10:24:06 AM) Total Marks: 1
Consider the code below, class class1{ public: void func1(); }; class class2 : private class1 {}; Function func1 of class1 is _____ in class2,
Select correct option:

public

protected

private

none of the given options

Question # 7 of 10 (Start time: 10:25:15 AM) Total Marks: 1
Function overloading is done with respect of,
Select correct option:

A single class

A derived class

A base class

Both derived and base classes

p-203

Question # 8 of 10 (Start time: 10:25:44 AM) Total Marks: 1
Child class can call constructor of its,
Select correct option:

Direct base class

Indirect base class

Both direct and indirect base classes

None of these

Question # 9 of 10 (Start time: 10:26:22 AM) Total Marks: 1
Consider the code below, class class1{ private: void func1(); }; class class2 : public class1 {}; Function func1 of class1 is _____ in class2,
Select correct option:

public

protected

private

none of the given options

Question # 10 of 10 (Start time: 10:27:26 AM) Total Marks: 1

Sender of the message does not need to know the exact class of receiver in_____.
Select correct option:

Abstraction

Polymorphism

Inheritance

none of the given
p-225

==========>

MCQz (Set-7)

1 -Target of a _____ function call is determined at run time.
Select correct option:

instance

virtual

operator

none of given

2 A Child class can call constructor of its parent class through,
Select correct option:

Its constructor initialization list

Its constructor body

Both from its constructor initialization list or body

Can not call the constructor of its parent class

3 By default, assignment operator (=) performs,
Select correct option:

Shallow copy page no. 192

Deep copy

Both of these

None of these

4 Sender of the message does not need to know the exact class of receiver in_____.
Select correct option:

Abstraction

Polymorphism

Inheritance

none of the given

5 In c , compiler can generate which of the following operators' code,
Select correct option:

= =

=

&

&&

6 Virtual functions allow you to
Select correct option:

create an array of type pointer-to-base class that can hold pointers to derived classes.

create functions that can never be accessed.

***** objects of different classes so they can all be accessed by the same function code.

use the same function call to execute member functions of objects from different classes.

7 Two functions with same names, parameters and return type can exist in,
Select correct option:

Function overloading

Function overriding

Operator overloading

None of these options

8 Consider the code below, class c1{ }; class c2 : public c1 { }; class c3 : public c2 { }; Then c2 is,
Select correct option:

Direct base class of c3

Direct child class of c3

Direct base class of c1

None of these

In case of dynamic memory allocation in a class we should use,

Select correct option:

User defined default constructor

User defined copy constructor

Both of these

None of these
==========>

MCQz (Set-8)

of the following is TRUE,

Select correct option:

- Derived class pointer can be used as Base class pointer
- Base class pointer can be used as Derived class pointer
- Both of these options
- None of these options

Compiler generated copy constructor performs,

Select correct option:

- Shallow copy
- Deep copy
- Both of these options
- None of these options

Friends are used exactly the same for template and non-template classes.

Select correct option:

- True
- False

Consider the code below, class class1{ public: void func1(); }; class class2 : private class1 {}; Function func1 of class1 is _____ in class2,

Select correct option:

- public
- protected
- **private**
- none of the given options

Consider the following statements: 1) int iArray[5]; 2) int *pArr = iArray;

Select correct option:

- These statements will compile successfully
- Error in first statement
- Error in second statement
- None of given options

Consider the following two lines of code written for a class Student, 1. Student sobj1,sobj2; 2. sobj2 = sobj1; In line No.2 what constructor of Student class will be called,

Select correct option:

- Default constructor of Student class.
- Copy constructor of student class
- Both default and copy constructer of Student class
- No constructor will be called.

User can make virtual table explicitly.

Select correct option:

- True
- False

A template provides a convenient way to make a family of

Select correct option:

- variables and data members
- functions and classes
- classes and exceptions
- programs and algorithms

A class template may inherit from another class template.

Select correct option:

- True
- False

Consider the code below, class class1{ private: void func1(); }; class class2 : protected class1 {}; Function func1 of class1 is _____ in class2,

Select correct option:

- public
- protected
- private

- none of the given options

A template argument is preceded by the keyword _____.

Select correct option:

- vector
- class
- template
- type*

We can call base class constructor from derived class constructor,

Select correct option:

- From derived class constructor body
- From the initializer list of derived class constructor
- From any member function of derived class
- We can not call the base class constructor

A function template must have a parameter.

Select correct option:

- True
- False

Consider the code below, class class1{ protected: int i; }; class class2 : public class1 { }; Then int member i of class1 is _____ in class2,

Select correct option:

- public
- protected
- private
- none of the given options

When we want to implement one class in terms of another class then we use,

Select correct option:

- Public inheritance
- Protected inheritance
- Private inheritance
- None of these options

==========>

MCQz (Set-9)

Question No: 1 (Marks: 1) - Please choose one

A template provides a convenient way to make a family of

► **variables and data members**

► functions and classes

► classes and exceptions

► programs and algorithms

Question No: 2 (Marks: 1) - Please choose one

Which one of the following terms must relate to **polymorphism?**

► Static allocation

► Static typing

► **Dynamic binding**

► Dynamic allocation

Question No: 3 (Marks: 1) - Please choose one

What is true about function templates?

► The compiler generates only one copy of the function template

► **The compiler generates a copy of function respective to each type of data**

► The compiler can only generate copy for the int type data

► None of the given.

Question No: 4 (Marks: 1) - Please choose one

Which of the following is the best approach if it is required to have more than one functions having exactly same functionality and implemented on different data types?

► **Templates**

► Overloading

► Data hiding

► Encapsulation

Question No: 5 (Marks: 1) - Please choose one
template <>
class Vector<char*> {}

This is an example of partial specialization.
- ► True
- ► **False**

Question No: 6 (Marks: 1) - Please choose one
Classes like TwoDimensionalShape and ThreeDimensionalShape would normally be concrete, while classes like Sphere and Cube would normally be abstract.
- ► **True**
- ► False

Question No: 7 (Marks: 1) - Please choose one
A non-virtual member function is defined in a base class and overridden in a derived class; if that function is called through a base-class pointer to a derived class object, the derived-class version is used.

- ► **True**
- ► False

Question No: 8 (Marks: 1) - Please choose one
Assume a class Derv that is privately derived from class Base. An object of class Derv located in main() can access
- ► public members of Derv.
- ► **protected members of Derv.**
- ► private members of Derv.
- ► protected members of Base.

Question No: 9 (Marks: 1) - Please choose one
In order to define a class template, the first line of definition must be:
- ► **template <typename T>**
- ► typename <template T>
- ► Template Class <ClassName>
- ► Class <Template T>

Question No: 10 (Marks: 1) - Please choose one
If there is a pointer p to objects of a base class, and it contains the address of an object of a derived class, and both classes contain a nonvirtual member function, ding(), then the statement p->ding(); will cause the version of ding() in the _____ class to be executed.
- ► Base
- ► Derived
- ► Abstract
- ► virtual

Question No: 11 (Marks: 1) - Please choose one

When the base class and the derived class have a member function with the same name, you must be more specific which function you want to call (using _____).

- ► scope resolution operator
- ► dot operator
- ► **null operator**
- ► Operator overloading

Question No: 12 (Marks: 1) - Please choose one
Non Template Friend functions of a class are friends of _____instance/s of that class.

- ► All
- ► **One specific**
- ► All instances of one date type
- ► None of the given options

Question No: 13 (Marks: 1) - Please choose one
The find() algorithm

- ► finds matching sequences of elements in two containers.
- ► **finds a container that matches a specified container.**
- ► takes iterators as its first two arguments.
- ► takes container elements as its first two arguments.

Question No: 14 (Marks: 1) - Please choose one
If you define a vector v with the default constructor, and define another vector w with a one-argument constructor to a size of 11, and insert 3 elements into each of these vectors with push_back(), then the size() member function will return _____ for v and _____ for w.

- ► 11 for v and 3 for w.
- ► 0 for v and 0 for w.
- ► 0 for v and 3 for w.
- ► **3 for v and 11 for w.**

Question No: 15 (Marks: 1) - Please choose one
Which of the following may not be an integral part of an object?

- ► State
- ► Behavior
- ► Protected data members
- ► **All of given**

Question No: 16 (Marks: 1) - Please choose one
Which is not the Advantage of inheritance?

► providing class growth through natural selection.
► **facilitating class libraries.**
► avoiding the rewriting of code.
► providing a useful conceptual framework.

Question No: 17 (Marks: 1) - Please choose one

```
class DocElement
{
public:
    virtual void Print() { cout << "Generic element"; }
};
class Heading : public DocElement
{
public:
    void Print() { cout << "Heading element"; }
};
class Paragraph : public DocElement
{
public:
    void Print() { cout << "Paragraph element"; }
};
void main()
{
    DocElement * p = new Paragraph();

    p->Print();
}
```

When you run this program, it will print out a single line to the console output.

What will be in that line?

Select one correct answer from the following list:

► Generic element

► Heading element

► **Paragraph element**

► Nothing will be printed.

Question No: 18 (Marks: 1) - Please choose one

When a virtual function is called by referencing a specific object by name and using the dot member selection operator (e.g., squareObject.draw()), the reference is resolved at compile time.

► **True**
► False

Question No: 19 (Marks: 1) - Please choose one
In case of multiple inheritance a derived class inherits,
- ► Only the public member functions of its base classes
- ► Only the public data members of its base classes
- ► **Both public data members and member functions of all its base classes**
- ► Data members and member functions of any two base classes

Question No: 20 (Marks: 1) - Please choose one
When we write a class template the first line must be:
- ► template < class *class_name*>
- ► template < class *data_type*>
- ► **template < class T >**

Here T can be replaced with any name but it is preferable.
- ► class class-name()

class template<*class_name*>

Question No: 21 (Marks: 1) - Please choose one
Which of the following is incorrect line regarding function template?

- ► template<class T>
- ► template <typename U>
- ► **Class<template T>**
- ► template < class T, class U>

Question No: 22 (Marks: 1) - Please choose one
An STL container can not be used to,

- ► hold objects of class employee.
- ► store elements in a way that makes them quickly accessible.
- ► compile c++ programs.
- ► organize the way objects are stored in memory

Question No: 23 (Marks: 1) - Please choose one
Algorithms can only be implemented using STL containers.

- ► **True**
- ► False

Question No: 24 (Marks: 1) - Please choose one
Consider a class named Vehicle, which of the following can be the instance of class Vehicle?

1. Car
2. Computer
3. Desk
4. Ahmed
5. Bicycle
6. Truck

► 1, 4, 5

► 2, 5, 6

► 1, 2, 3, 6

► 1, 5, 6

Question No: 25 (Marks: 1) - Please choose one

Consider the code below,

```
class Fred {
public:
Fred();
…
};
int main()
{
Fred a[10];
Fred* p = new Fred[10];
…
}
```

Select the best option,

► Fred a[10]; calls the default constructor 09 times
Fred* p = new Fred[10]; calls the default constructor 10 times

► Produce an error

► Fred a[10]; calls the default constructor 11 times
Fred* p = new Fred[10]; calls the default constructor 11 times

► Fred a[10]; calls the default constructor 10 times
Fred* p = new Fred[10]; calls the default constructor 10 times

Question No: 26 (Marks: 1) - Please choose one

When a variable is define as **static** in a class then all object of this class,

► Have different copies of this variable
► Have same copy of this variable
► Can not access this variable
► None of given

Question No: 27 (Marks: 1) - Please choose one
The life of sub object is dependant on the life of master class in _____.

- ► Separation
- ► **Composition**
- ► Aggregation
- ► None of the given

Question No: 28 (Marks: 1) - Please choose one
_____, which means if A declares B as its friend it does NOT mean that A can access private data of B. It only means that B can access all data of A.

- ► Friendship is one way only
- ► Friendship is two way only
- ► NO Friendship between classes
- ► Any kind of friendship

Question No: 29 (Marks: 1) - Please choose one
Which of the following operators always takes no argument if overloaded?

- ► /
- ► -
- ► +
- ► **++**

Question No: 30 (Marks: 1) - Please choose one
In Private ---------- only member functions and friend classes or functions of a derived class can convert pointer or reference of derived object to that of parent object

- ► specialization
- ► inheritance
- ► abstraction
- ► composition

==========>

MCQz (Set-10)

Question No: 1 (Marks: 1) - Please choose one

A template argument is preceded by the keyword _____.

- ► **vector**
- ► **class**
- ► **template**
- ► **type***

Question No: 2 (Marks: 1) - Please choose one

Which of the following causes run time binding?

► Declaring object of abstract class

► Declaring pointer of abstract class

► Declaring overridden methods as non-virtual

► None of the given

Question No: 3 (Marks: 1) - Please choose one

A function template can not be overloaded by another function template.

► True

► False

Question No: 4 (Marks: 1) - Please choose one

Which of the following is the best approach if it is required to have more than one functions having exactly same functionality and implemented on different data types?

► Templates

► Overloading

► Data hiding

► Encapsulation

Question No: 5 (Marks: 1) - Please choose one

Identify the correct way of declaring an object of user defined template class A for char type members?

► A< char > obj;

► <char>A obj;

► A obj<char>;

► Obj <char> A;

Question No: 6 (Marks: 1) - Please choose one

The user must define the operation of the copy constructor.

► True

► False

Question No: 7 (Marks: 1) - Please choose one
Template functions use _____ than ordinary functions.

► Greater Memory

► Lesser Memory

► Equal Memory

► None of the given options

Question No: 8 (Marks: 1) - Please choose one
The find() algorithm

► finds matching sequences of elements in two containers.

► finds a container that matches a specified container.

► takes iterators as its first two arguments.

► takes container elements as its first two arguments.

Question No: 9 (Marks: 1) - Please choose one
Compiler performs _____ type checking to diagnose type errors,

► Static

► Dynamic

► Bound

► Unbound

Question No: 10 (Marks: 1) - Please choose one
Which of the following is/are advantage[s] of generic programming?

► Reusability

► Writability

► Maintainability

► All of given

Question No: 11 (Marks: 1) - Please choose one
Vectors contain contiguous elements stored as a[an] ___.

► variable

► array

► function

► datatype

Question No: 12 (Marks: 1) - Please choose one
Suppose you create an uninitialized vector as follows:

vector<int> evec;
After adding the statment,
evec.push_back(21);
what will happen?

► The following statement will add an element to the start (the back) of evec and will initialize it with the value 21.

► The following statement will add an element to the center of evec and will reinitialize it with the value 21.

► The following statement will delete an element to the end (the back) of evec and will reinitialize it with the value 21.

► The following statement will add an element to the end (the back) of evec and initialize it with the value 21.

Question No: 13 (Marks: 1) - Please choose one
In a de-queue, (chose the best option)

► data can be quickly inserted or deleted at any arbitrary location.

► data can be inserted or deleted at any arbitrary location, but the process is relatively slow.

► data can not be quickly inserted or deleted at either end.

► data can be inserted or deleted at either end, but the process is relatively slow.

Question No: 14 (Marks: 1) - Please choose one
Algorithms can only be implemented using STL containers.

► True

► False

Question No: 17 (Marks: 1) - Please choose one
We can use "this" pointer in the constructor in the body and even in the initialization list of any class if we are careful,

► True

► False

Question No: 18 (Marks: 1) - Please choose one

► private,static

► private,public

► static,public

► none of given

Question No: 19 (Marks: 1) - Please choose one
Default constructor is such constructor which either has no ---------or if it has some parameters these have -------- values

► Parameter, temporary

► Null, Parameter

► Parameter, default

► non of the given

Question No: 20 (Marks: 1) - Please choose one

Public methods of base class can --------- be accessed in its derived class

► directly

► inderectly

► simultaniously

► non of the given

Question No: 21 (Marks: 1) - Please choose one
The type that is used to declare a reference or pointer is called its ---------

► default type

► static type

► abstract type

► reference type

Question No: 22 (Marks: 1) - Please choose one
---------- members are somewhere between public and private members. They are used in inheritance

► protected

► public

► **private**

► **global**

Question No: 23 (Marks: 1) - Please choose one
Which of these are examples of error handling techniques ?

► **Abnormal Termination**

► **Graceful Termination**

► **Return the illegal**

► **all of the given**

Question No: 24 (Marks: 1) - Please choose one
---------- follow try block to catch the object thrown

► **catch block**

► **throw block**

► **main block**

► **non of the given**

Question No: 25 (Marks: 1) - Please choose one
Graphical representation of the classes and objects is called object model it shows -
------

► **Class Name only**

► **Class Name and attributes**

► **Relationships of the objects and classes**

► **all of the given**

Question No: 26 (Marks: 1) - Please choose one
Destructor can be overloaded

► **True**

► **False**

==========>

MCQz (Set-11)

Question # 1 of 10 (Start time: 08:40:52 AM) Total Marks: 1
Outside world can access only _____ members of a class using its object.
Select correct option:

Public
Private
Protected

No member is accessible.

Question # 2 of 10 (Start time: 08:41:55 AM) Total Marks: 1
Consider the code below, class class1{ private: int i; }; class class2 : protected class1 {};
Then int member i of class1 is _____ in class2,
Select correct option:

public
protected
private
none of the given options

Question # 4 of 10 (Start time: 08:44:20 AM) Total Marks: 1
In Public Inheritance the public members of base class become _____ in derived class.
Select correct option:

Public
Private
Protected
None of the given options.

Question # 5 of 10 (Start time: 08:45:35 AM) Total Marks: 1
Consider the code below, class class1{ protected: int i; }; class class2 : public class1 {};
Then int member i of class1 is _____ in class2,
Select correct option:

public
protected
private
none of the given options

Question # 7 of 10 (Start time: 08:47:08 AM) Total Marks: 1
Consider the following two lines of code written for a class Student, 1. Student sobj1; 2. Student sobj2(sobj1); In line No.2 what constructor of Student class will be called,
Select correct option:

Default constructor of Student class
Copy constructor of student class
Both default and copy constructer of Student class
No constructor will be called

Question # 8 of 10 (Start time: 08:48:08 AM) Total Marks: 1
Consider the code below, class class1{ public: void func1(); }; class class2 : public class1 { }; Function func1 of class1 is _____ in class2,
Select correct option:

public
protected

private
none of the given options

Question # 9 of 10 (Start time: 08:49:18 AM) Total Marks: 1
It is illegal to make objects of one class members of another class.
Select correct option:

True
False

Question # 10 of 10 (Start time: 08:50:33 AM) Total Marks: 1
Which one of the following operators is used to access members of value data types,
Select correct option:

->
.
&
None of the given options

Question # 1 of 10 (Start time: 08:52:57 AM) Total Marks: 1
Static casting is,
Select correct option:

C++ way of calling base class functions from derived class
C way of calling base class functions from derived class
Both of these
None of these

Question # 2 of 10 (Start time: 08:53:33 AM) Total Marks: 1
_____ Binding means that target function for a call is selected at compile time.
Select correct option:

Static
Dynamic
Automatic
None of given

Question # 6 of 10 (Start time: 08:54:59 AM) Total Marks: 1
When we want to implement one class in terms of another class then we use,
Select correct option:

Public inheritance
Protected inheritance
Private inheritance
None of these options

Question # 10 of 10 (Start time: 08:57:13 AM) Total Marks: 1

Consider the code below, class class1{ protected: void func1(); }; class class2 : private class1 {}; Function func1 of class1 is _____ in class2,
Select correct option:

public
protected
private
none of the given options

Question # 1 of 10 (Start time: 08:59:16 AM) Total Marks: 1
Static casting is,
Select correct option:

Implicit way of calling base class functions from derived class
Explicit way of calling base class functions from derived class
None of these
Both of these

Question # 2 of 10 (Start time: 09:00:44 AM) Total Marks: 1
The default inheritance mode is,
Select correct option:

Public inheritance
Protected inheritance
Private inheritance
None of these options

Question # 3 of 10 (Start time: 09:01:47 AM) Total Marks: 1
In Private Inheritance the public members of base class become _____ in derived class.
Select correct option:

Public
Private
Protected
None of the given options.

Question # 4 of 10 (Start time: 09:02:49 AM) Total Marks: 1
Consider the code below, class class1{ private: void func1(); }; class class2 : public class1 {}; Function func1 of class1 is _____ in class2,
Select correct option:

public
protected
private
none of the given options

Question # 6 of 10 (Start time: 09:05:16 AM) Total Marks: 1
Function overloading is done with respect of,

Select correct option:

A single class
A derived class
A base class
Both derived and base classes

Question # 7 of 10 (Start time: 09:06:14 AM) Total Marks: 1
_____ Binding means that target function for a call is selected at run time
Select correct option:

Automatic
Dynamic
Static
Dramatic

Question # 8 of 10 (Start time: 09:06:39 AM) Total Marks: 1
In Protected Inheritance the public members of base class become _____ in derived class.
Select correct option:

Public
Private
Protected
None of the given options.

Question # 9 of 10 (Start time: 09:07:45 AM) Total Marks: 1
Consider the code below, class class1{ private: void func1(); }; class class2 : protected class1 {}; Function func1 of class1 is _____ in class2,
Select correct option:

public
protected
private
none of the given options

Question # 10 of 10 (Start time: 09:08:40 AM) Total Marks: 1
Suppose both derive and base classes have compiler generated copy constructors then derived class copy constructor will call base class _____,
Select correct option:

Compiler generated default constructor
Compiler generated copy constructor
Both of these options are correct
None of these options are correct

Question # 1 of 10 (Start time: 09:11:39 AM) Total Marks: 1
A class D can be derived from a class C, which is derived from a class B, which is derived from a class A.

Select correct option:

True
False

Question # 2 of 10 (Start time: 09:12:35 AM) Total Marks: 1
Consider the code below, class c1{}; class c2 : public c1 {}; class c3 : public c2 {}; Then c3 is,
Select correct option:

Direct base class of c1
Direct child class of c1
Direct base class of c2
Direct child class of c2

Question # 7 of 10 (Start time: 09:15:12 AM) Total Marks: 1
In specialization we can,
Select correct option:

Replace child class with its base class
Replace base class with its child class
Replace both child and base classes interchangeably
None of the given options

Question # 9 of 10 (Start time: 09:17:59 AM) Total Marks: 1
In private inheritance derived class pointer can be assigned to base class pointer in,
Select correct option:

Main function
In derived class member and friend functions
In base class member and friend functions
None of the given options
==========>

MCQz (Set-12)

Question No: 1 (Marks: 1) - Please choose one
Which one of the following terms must relate to **polymorphism?**
► Static allocation
► Static typing
► **Dynamic binding**
► Dynamic allocation

Question No: 2 (Marks: 1) - Please choose one
Multiple inheritance can be of type
► Public
► Private
► Protected
► **All of the given**

Question No: 3 (Marks: 1) - Please choose one
When a subclass specifies an alternative definition for an attribute or method of its superclass, it is _____ the definition in the superclass.

► overload
► **overriding**
► copy riding
► none of given

Question No: 4 (Marks: 1) - Please choose one
Like template functions, a class template may not handle all the types successfully.
► **True**
► False

Question No: 5 (Marks: 1) - Please choose one
It is sometimes useful to specify a class from which no objects will ever be created.
► True
► **False**

Question No: 6 (Marks: 1) - Please choose one
Assume a class Derv that is privately derived from class Base. An object of class Derv located in main() can access
► public members of Derv.
► protected members of Derv.
► private members of Derv.
► **protected members of Base.**

Question No: 7 (Marks: 1) - Please choose one
A pointer to a base class can point to objects of a derived class.
► **True**
► False

Question No: 8 (Marks: 1) - Please choose one
A copy constructor is invoked when
► a function do not returns by value.
► an argument is passed by value.
► **a function returns by reference.**
► an argument is passed by reference.

Question No: 9 (Marks: 1) - Please choose one
A function call is resolved at run-time in_____
► non-virtual member function.
► **virtual member function.**
► Both non-virtual member and virtual member function.
► None of given

Question No: 10 (Marks: 1) - Please choose one
When the base class and the derived class have a member function with the same name, you must be more specific which function you want to call (using _____).
► **scope resolution operator**
► dot operator
► null operator
► Operator overloading

Question No: 11 (Marks: 1) - Please choose one
Each try block can have _____ no. of catch blocks.
► 1
► 2
► 3
► **As many as necessary.**

Question No: 12 (Marks: 1) - Please choose one

Two important STL associative containers are _____ and _____.
► set,map
► **sequence,mapping**
► setmet,multipule
► sit,mat

Question No: 13 (Marks: 1) - Please choose one
The mechanism of selecting function at run time according to the nature of calling object is called,
► late binding
► static binding
► virtual binding
► **None of the given options**

Question No: 14 (Marks: 1) - Please choose one
An abstract class is useful when,
► We do not derive any class from it.
► There are multiple paths from one derived class to another.
► **We do not want to instantiate its object.**
► You want to defer the declaration of the class.

Question No: 15 (Marks: 1) - Please choose one
Which of the following is incorrect line regarding function template?
► template<class T>
► template <typename U>
► **Class<template T>**
► template < class T, class U>

Question No: 16 (Marks: 1) - Please choose one
Which of the following is/are advantage[s] of generic programming?
► Reusability
► Writability
► Maintainability
► **All of given**

Question No: 17 (Marks: 1) - Please choose one
By default the vector data items are initialized to ____
► **0**
► 0.0
► 1
► null

Question No: 18 (Marks: 1) - Please choose one
Which one of the following functions returns the total number of elements in a vector.
► length();
► **size();**
► ele();
► veclen();

Question No: 19 (Marks: 1) - Please choose one
Suppose you create an uninitialized vector as follows:
vector<int> evec;
After adding the statment,
evec.push_back(21);
what will happen?
► The following statement will add an element to the start (the back) of evec and will

initialize it with the value 21.
► The following statement will add an element to the center of evec and will reinitialize it with the value 21.
► The following statement will delete an element to the end (the back) of evec and will reinitialize it with the value 21.
► The following statement will add an element to the end (the back) of evec and initialize it with the value 21.

Question No: 20 (Marks: 1) - Please choose one
An STL container can not be used to,
► hold objects of class employee.
► store elements in a way that makes them quickly accessible.
► compile c++ programs.
► organize the way objects are stored in memory

Question No: 21 (Marks: 1) - Please choose one
Algorithms can only be implemented using STL containers.
► True
► False

Question No: 22 (Marks: 1) - Please choose one
The main function of scope resolution operator (**::**) is,
► To define an object
►To define a data member
►To link the definition of an identifier to its declaration
►To make a class private

Question No: 23 (Marks: 1) - Please choose one
When is a constructor called?
►Each time the constructor identifier is used in a program statement
► During the instantiation of a new object
►During the construction of a new class
►At the beginning of any program execution

Question No: 24 (Marks: 1) - Please choose one
Consider the code below,

```
class Fred {
public:
Fred();
…
};
int main()
{
Fred a[10];
Fred* p = new Fred[10];
…
}
```

Select the best option,
►Fred a[10]; calls the default constructor 09 times
Fred* p = new Fred[10]; calls the default constructor 10 times
► Produce an error
►Fred a[10]; calls the default constructor 11 times
Fred* p = new Fred[10]; calls the default constructor 11 times
►Fred a[10]; calls the default constructor 10 times

Fred* p = new Fred[10]; calls the default constructor 10 times

Question No: 25 (Marks: 1) - Please choose one

Associativity can be changed in operator overloading.

►True

► False

Question No: 26 (Marks: 1) - Please choose one

A normal C++ operator that acts in special ways on newly defined data types is said to be

►glorified.

►encapsulated.

► classified.

►overloaded.

Question No: 27 (Marks: 1) - Please choose one

Which operator can not be overloaded?

►The relation operator (>=)

►Assignment operator (=)

►Script operator ([])

► Conditional operator (? :)

Question No: 28 (Marks: 1) - Please choose one

Suppose obj1 and obj2 are two objects of a user defined class A. An + operator is overloaded to add obj1 and obj2 using the function call obj1+obj2.

Identify the correct function prototype against the given call?

► A operator + (A &obj);

► int + operator();

► int operator (plus) ();

► A operator(A &obj3);

Question No: 29 (Marks: 1) - Please choose one

Default constructor is such constructor which either has no ---------or if it has some parameters these have -------- values

► Parameter, temporary

► Null, Parameter

► Parameter, default

► non of the given

Question No: 30 (Marks: 1) - Please choose one

Public methods of base class can --------- be accessed in its derived class

► directly

► **inderectly**

► simultaniously

► non of the given

Question No: 31 (Marks: 1)

Is **Deque** a Birectional Container?

Yes, deque behaves like queue (line) such that we can add elements on both sides of it.

Question No: 32 (Marks: 1)

What is meant by Generic Programming?

Generic programming refers to programs containing generic abstractions general code that is same in logic for all data types like printArray function), then we instantiate that generic program abstraction (function, class) for a particular data type, such abstractions can work with many different types of data.

==========>

MCQz (Set-13)

Question # 1 of 10
Information hiding can be achieved through_____.
1.Encapsulation, Inheritance
2.Encapsulation, Polymorphism
3.Encapsulation, Abstraction
4.Overloading
Question # 2 of 10 (Start time: 01:11:21 AM) Total Marks: 1
A good model is ..... related to a real life problem.
Select correct option:
1.Loosely
2.Openly
3.Closely
Question # 3 of 10 (Start time: 01:12:33 AM) Total Marks: 1
Which of the following features of OOP is used to derive a class from another?
Select correct option:
1.Encapsulation
2.Polymorphism
3.Data hiding
4. Inheritance
Question # 4 of 10 (Start time: 01:13:51 AM) Total Marks: 1
Which of the following is a weak relationship between two objects?
Select correct option:
1. Inheritance
2.Composition
3.Aggregation
4.None of given
Question # 5 of 10 (Start time: 01:14:56 AM) Total Marks: 1
Data items in a class must be private.
Select correct option:
1.True
2.False
Question # 7 of 10 (Start time: 01:16:55 AM) Total Marks: 1
Suppose there is an object of type Person, which of the following can be considered as one of its attributes
Select correct option:
1.Name
2.Age
3.Work()
4.Both Name and Age
Question # 8 of 10 (Start time: 01:17:52 AM) Total Marks: 1
Which one is not an object association?
Select correct option:
1.Simple association
2. Inheritance
3.Aggregation
4.Association

Question # 9 of 10 (Start time: 01:18:50 AM) Total Marks: 1
Using encapsulation we can achieve
Select correct option:
1. Information hiding
2.Least interdependencies among modules
3. Implementation independence
4.All of given options
Question # 10 of 10 (Start time: 01:19:43 AM) Total Marks: 1
In constant member function the type of this pointer is:
Select correct option:
1.Constant pointer
2.Constant pointer to object
3.Constant pointer to class
4.Constant pointer to constant object
Question # 1 of 10
Which of the following is the way to extract common behavior and attributes from the given classes and make a separate class of those common behaviors and attributes?
1.Generalization
2.Sub-typing
3.Specialization
4.Extension
Question # 2 of 10
The ability to derive a class from more than one class is called
1.Single inheritance
2.Encapsulation
3.Multiple inheritance
4.Polymorphism
Question # 3 of 10:
If MyClass has a destructor what is the destructor named?
1.MyClass
2.~MyClass
3.My~Class
4.MyClass~
Question # 4 of 10:
Class abc{ ----- }; Is a valid class declaration?
1.yes
2.no
Question # 5of 10:
Without using Deep copy constructor, A _____ problem can occur
1.System crash
2.Memory Leakage
3.Dangling pointer
4.All of the given
Question # 6 of 10:
If only one behaviour of a derived class is incompatible with base class, then it is:
1.Generalization

2.Specialization

3.Extension

4. Inheritance

Question # 7 of 10:

Which of the following may not be an integral part of an object?

1. state

2.behavior

3.Protected data members

4.All of given

Question # 8 of 10:

Only tangible things can be chosen as an object.

1.True

2.False

1.

2. Question # 1 of 10

Information hiding can be achieved through_____.

Encapsulation, Inheritance

Encapsulation, Polymorphism

Encapsulation, Abstraction

Overloading

Question # 2 of 10 (Start time: 01:11:21 AM) Total Marks: 1

A good model is ..... related to a real life problem.

Select correct option:

Loosely

Openly

Closely

Question # 3 of 10 (Start time: 01:12:33 AM) Total Marks: 1

Which of the following features of OOP is used to derive a class from another?

Select correct option:

Encapsulation

Polymorphism

Data hiding

Inheritance

Question # 4 of 10 (Start time: 01:13:51 AM) Total Marks: 1

Which of the following is a weak relationship between two objects?

Select correct option:

Inheritance

Composition

Aggregation

None of given

Question # 5 of 10 (Start time: 01:14:56 AM) Total Marks: 1

Data items in a class must be private.

Select correct option:

True

False

Question # 7 of 10 (Start time: 01:16:55 AM) Total Marks: 1

Suppose there is an object of type Person, which of the following can be considered as one of its attributes

Select correct option:

Name
Age
Work()

Both Name and Age

Question # 8 of 10 (Start time: 01:17:52 AM) Total Marks: 1
Which one is not an object association?
Select correct option:
Simple association

Inheritance

Aggregation
Association
Question # 9 of 10 (Start time: 01:18:50 AM) Total Marks: 1
Using encapsulation we can achieve
Select correct option:

Information hiding

Least interdependencies among modules
Implementation independence
All of given options
Question # 10 of 10 (Start time: 01:19:43 AM) Total Marks: 1
In constant member function the type of this pointer is:
Select correct option:
Constant pointer

Constant pointer to object

Constant pointer to class
Constant pointer to constant object
Question # 1 of 10
Which of the following is the way to extract common behavior and attributes from the given classes and make a separate class of those common behaviors and attributes?

Generalization

Sub-typing
Specialization
Extension
Question # 2 of 10
The ability to derive a class from more than one class is called
Single inheritance
Encapsulation

Multiple inheritance

Polymorphism
Question # 3 of 10:
If MyClass has a destructor what is the destructor named?
MyClass

~MyClass

My~Class
MyClass~
Question # 4 of 10:
Class abc{ ----- }; Is a valid class declaration?

yes

no
Question # 5of 10:

Without using Deep copy constructor, A _____ problem can occur
System crash
Memory Leakage
Dangling pointer
All of the given
Question # 6 of 10:
If only one behaviour of a derived class is incompatible with base class, then it is:
Generalization
Specialization
Extension
Inheritance
Question # 7 of 10:
Which of the following may not be an integral part of an object?
state
behavior
Protected data members
All of given
Question # 8 of 10:
Only tangible things can be chosen as an object.
True
False
Class is not a mechanism to create objects and define user data types.
1. true
2. false

==========>

MCQz (Set-14)

Memory is allocated to non static members only, when:
1. Class is created
2. Object is defined
3. Object is initialized
4. Object is created
The sub-object's life is not dependent on the life of master class in _____.
1. Composition
2. Aggregation
3. Separation
4. non of the given
Unary operators and assignment operator are right associative.
1. true
2. false
The >= operator can't be overloaded.
1. true
2. false
_____ is creating objects of one class inside another class.
1. Association
2. Composition
3. Aggregation
4. Inheritance

If we are create array of objects through new operator, then
1. We can call overloaded constructor through new
2. We can't call overloaded constructor through new
3. We can call default constructor through new
4. None of the given
Object can be declared constant with the use of Constant keyword.
1. true
2. false
_____ Operator will take only one operand.
1. New
2. int
3. object
4. none of the given
Which of the following operator(s) take(s) one or no argument if overloaded?
1. ++
2. *
3. %
4. All of the given choices
this pointer does not pass implicitly to _____ functions.
1. **Static Member**
2. Non-Static Member
3. Instance Number
4. None of the given
Operator overloading is
1. making C++ operators work with objects.
2. giving C++ operators more than they can handle.
3. giving new meanings to existing Class members.
4. making new C++ operators

==========>

MCQz (Set-15)

Question # 1 of 10 (Start time: 09:57:41 AM) Total Marks: 1
Consider the code below, class class1{ public: void func1(); }; class class2 : private class1 {}; Function func1 of class1 is
_____ in class2,
Select correct option:
public
protected
private
none of the given options
Click here to Save Answer & Move to Next Question
Question # 2 of 10 (Start time: 09:59:01 AM) Total Marks: 1
User can make virtual table explicitly.
Select correct option:
True
False
Click here to Save Answer & Move to Next Question
Question # 3 of 10 (Start time: 10:00:15 AM) Total Marks: 1
In private inheritance derived class pointer can be assigned to base class pointer in,

Select correct option:
Main function
In derived class member and friend functions
In base class member and friend functions
None of the given options
Click here to Save Answer & Move to Next Question
Question # 4 of 10 (Start time: 10:01:15 AM) Total Marks: 1
In C++, we declare a function virtual by preceding the function header with keyword "Inline"
Select correct option:
True
False
Click here to Save Answer & Move to Next Question
Question # 5 of 10 (Start time: 10:02:45 AM) Total Marks: 1
Outside world can access only _____ members of a class using its object.
Select correct option:
Public
Private
Protected
No member is accessible.
Click here to Save Answer & Move to Next Question
Question # 6 of 10 (Start time: 10:03:10 AM) Total Marks: 1
Friend Functions of a class are _____ members of that class.
Select correct option:
Public
Private
Protected
None of the given options.
Click here to Save Answer & Move to Next Question
Question # 7 of 10 (Start time: 10:03:54 AM) Total Marks: 1
Consider the following two lines of code written for a class Student, 1. Student sobj1,sobj2; 2. sobj2 = sobj1; In line No.2 what constructor of Student class will be called,
Select correct option:
Default constructor of Student class.
Copy constructor of student class
Both default and copy constructer of Student class
No constructor will be called.
Click here to Save Answer & Move to Next Question
Time Left 87 Class is not a mechanism to create objects and define user data types.
1. true
2. false
Memory is allocated to non static members only, when:
1. Class is created
2. Object is defined
3. Object is initialized
4. Object is created
The sub-object's life is not dependent on the life of master class in _____.
1. Composition

2. Aggregation
3. Separation
4. non of the given

Unary operators and assignment operator are right associative.
1. true
2. false

The >= operator can't be overloaded.
1. true
2. false

_____ is creating objects of one class inside another class.
1. Association
2. Composition
3. Aggregation
4. Inheritance

If we are create array of objects through new operator, then
1. We can call overloaded constructor through new
2. We can't call overloaded constructor through new
3. We can call default constructor through new
4. None of the given

Object can be declared constant with the use of Constant keyword.
1. true
2. false

_____ Operator will take only one operand.
1. New
2. int
3. object
4. none of the given

Which of the following operator(s) take(s) one or no argument if overloaded?
1. ++
2. *
3. %
4. All of the given choices

this pointer does not pass implicitly to _____ functions.
1. Static Member
2. Non-Static Member
3. Instance Number
4. None of the given

Operator overloading is
1. making C++ operators work with objects.
2. giving C++ operators more than they can handle.
3. giving new meanings to existing Class members.
4. making new C++ operators

Question # 1 of 8 (Start time: 10:39:47 PM) Total Marks: 1
Which of the following operator(s) take(s) one or no argument if overloaded?
Select correct option:
++
*
%

All of the given choices
Click here to Save Answ er & Move to Next Question

Question # 2 of 8 (Start time: 10:40:38 PM) Total Marks: 1
Object can be declared constant with the use of Constant keyword.
Select correct option:
True
False
Click here to Save Answ er & Move to Next Question

Question # 3 of 8 (Start time: 10:41:41 PM) Total Marks: 1
Static data members are called _____ variable
Select correct option:
Class
Object
Structure
none of the given
Click here to Save Answ er & Move to Next Question

Question # 4 of 8 (Start time: 10:42:35 PM) Total Marks: 1
Associatively can be change in operator overloading.
Select correct option:
True
False

Question # 5 of 8 (Start time: 10:43:56 PM) Total Marks: 1
_____ and _____ methods may not be declared abstract.
Select correct option:
Private,static
private,public
static,public
none of the given

Question # 6 of 8 (Start time: 10:45:17 PM) Total Marks: 1
Let Suppose a class Student with objects std1, std2, and std3. For the statement std3 = std1 - std2 to work correctly, if the overloaded - operator must
Select correct option:
take two arguments.
None of the given choices
take single argument
take three arguments

Question # 7 of 8 (Start time: 10:46:48 PM) Total Marks: 1
To initialize an array of objects, only _____ will be called
Select correct option:
Defualt Constructor
Overloaded Constructor
Default Object
None of the above

Question # 8 of 8 (Start time: 10:47:49 PM) Total Marks: 1
_____ provide the facility to access the data member.
Select correct option:
accesser function
private function
inline function
None of the given

Question # 8 of 10 (Start time: 10:04:41 AM) Total Marks: 1
Consider the following two lines of code written for a class Student, 1. Student sobj1; 2. Student sobj2 = sobj1; In line No.1
what constructor of student class will be called,
Select correct option:
Default constructor of Student class.
Copy constructor of student class
Both default and copy constructer of student class
None the given options

Question # 9 of 10 (Start time: 10:05:09 AM) Total Marks: 1
Consider the code below, class class1{ protected: void func1(); }; class class2 : protected class1 {}; Function func1 of class1 is
_____ in class2,
Select correct option:
public
protected
private
none of the given options

Question # 10 of 10 (Start time: 10:05:50 AM) Total Marks: 1
Virtual functions allow you to
Select correct option:
create an array of type pointer-to-base class that can hold pointers to derived classes.
create functions that can never be accessed.
group objects of different classes so they can all be accessed by the same function code.
use the same function call to execute member functions of objects from different classes.
==========>

MCQz (Set-16)

Question # 1 of 10 (Start time: 12:40:20 PM) Total Marks: 1
Which of the following is the way to extract common behaviour and attributes from the given classes and make a separate class of those common behaviours and attributes?
Select correct option:
Generalization
Sub-typing
Specialization
Extension

Question # 2 of 10 (Start time: 12:41:52 PM) Total Marks: 1
“A fan has wings”. Which type of relation exists between fan and wings in this sentence?

Select correct option:
Aggregation
Association
Generalization
Composition

Question # 3 of 10 (Start time: 12:42:46 PM) Total Marks: 1
A good model is ..... related to a real life problem.
Select correct option:
Loosely
Openly
Closely
Not

Question # 5 of 10 (Start time: 12:44:45 PM) Total Marks: 1
When we create objects, then space is allocated to:
Select correct option:
Member functions
Access specifier
Data members
None of the given

Question # 6 of 10 (Start time: 12:45:21 PM) Total Marks: 1
There is only one form of copy constructor.
Select correct option:
True
False

Question # 7 of 10 (Start time: 12:45:38 PM) Total Marks: 1
Which of the following features of OOP is used to deal with only relevant details?
Select correct option:
Abstraction
Information hiding
Object
Inheritance

Question # 8 of 10 (Start time: 12:48:26 PM) Total Marks: 1
Suppose there is an object of type Person, which of the following can be considered as one of its attributes
Select correct option:
Age
Work()
Both Name and Age

Question # 9 of 10 (Start time: 12:56:04 PM) Total Marks: 1
Through interface we access object_____.
Select correct option:
States
Data members

Behaviour
None of the given

Question # 10 of 10 (Start time: 12:57:00 PM) Total Marks: 1
If a class A inherits from class B, then class A is called.
Select correct option:
Child class
Derived class
Parent class
Child and derived class

Question # 1 of 10 (Start time: 12:59:51 PM) Total Marks: 1
If some of objects exhibit identical characteristics, then they belong to:
Select correct option:
Different classes
Multiple classes
Same class
None of the given

Question # 2 of 10 (Start time: 01:00:41 PM) Total Marks: 1
_____ is automatically called when the object is created.
Select correct option:
member function
object
constructor
None of the given

Question # 3 of 10 (Start time: 01:03:09 PM) Total Marks: 1
Which is true about sub-typing in case of inheritance?
Select correct option:
In sub-typing a new class is derived from existing w
extended behavior of its parent.
In sub-typing a new class is derived from existing w
In sub-typing a class is derived from existing one wh
None of the given.

Question # 4 of 10 (Start time: 01:04:28 PM) Total Marks: 1
If a class involves dynamic memory allocation, then:
Select correct option:
Default copy constructor, shallow copy is implement
User defined copy constructor, shallow copy is impl
Default copy constructor, deep copy is implemented
User defined copy constructor, deep copy is implem

Question # 5 of 10 (Start time: 01:05:37 PM) Total Marks: 1
Which one is a not class association
Select correct option:
Simple Association
Inheritance

Composition
Aggregation

Question # 6 of 10 (Start time: 01:06:50 PM) Total Marks: 1
Data items in a class must be private.
Select correct option:
True
False

Question # 7 of 10 (Start time: 01:07:16 PM) Total Marks: 1
Three main characteristics of "Object Oriented programming" are,
Select correct option:
Encapsulation,dynamic binding,polymarhpishm
polymorphism, overloading, overriding
encapsulation, inheritance, dynamic binding
encapsulation, inheritance, polymorphism

Question # 8 of 10 (Start time: 01:08:14 PM) Total Marks: 1
Which of the following is the way to extract common behaviour and attributes from the given classes and make a separate class of those common behaviours and attributes?
Select correct option:
Generalization
Sub-typing
Specialization
Extension

Question # 9 of 10 (Start time: 01:09:04 PM) Total Marks: 1
The sentence “Object Oriented Programming book in bookshelf” is an example of:
Select correct option:
Association
Multiple association
Aggregation

Question # 10 of 10 (Start time: 01:16:05 PM) Total Marks: 1
Data members are the attributes of objects.
Select correct option:
True
False

Question # 1 of 10 (Start time: 01:18:48 PM) Total Marks: 1
Constructor have same name as the class name.
Select correct option:
True
False

Question # 2 of 10 (Start time: 01:19:03 PM) Total Marks: 1
Which of the following features of OOP is used to derive a class from another?
Select correct option:
Encapsulation

Polymorphism
Data hiding
Inheritance

Question # 3 of 10 (Start time: 01:19:29 PM) Total Marks: 1
Class abc{ ----- }; Is a valid class declaration?
Select correct option:
Yes
No

Question # 6 of 10 (Start time: 01:22:47 PM) Total Marks: 1
Which of the following is a weak relationship between two objects?
Select correct option:
Inheritance
Composition
Aggregation
None of given

Question # 4 of 10 (Start time: 01:20:47 PM) Total Marks: 1
Without using Deep copy constructor, A _____ problem can occur
Select correct option:
System crash
Memory Leakage
Dangling pointer
All of the given

Question # 5 of 10 (Start time: 01:21:20 PM) Total Marks: 1
An abstract class shows _____ behaviour.
Select correct option:
Overriding
Specific
General
None of the given

Question # 8 of 10 (Start time: 01:24:19 PM) Total Marks: 1
If a class A inherits from class B, then class A is called.
Select correct option:
Child class
Derived class
Parent class
Child and derived class

Question # 9 of 10 (Start time: 01:24:44 PM) Total Marks: 1
Consider the statement “room has chair” Which of the following type of association exists between room and chair?
Select correct option:
Inheritance
Composition
There is no association

Aggregation

Question # 10 of 10 (Start time: 01:25:05 PM) Total Marks: 1
The dot operator (or class member access operator) connects the following two entities (reading from left to right):
Select correct option:
A class member and a class object
A class object and a class
A class and a member of that class
A class object and a member of that class
==========>

MCQz (Set-17)

Question No: 1 (Marks: 1) – Please choose one
Which part of an object exhibits its state?
► **Data**
► Operations
► Any public part
► Any private part
Question No: 2 (Marks: 1) – Please choose one
Inheritance is a way to
► organize data.
► pass arguments to objects of classes.
► **add features to existing classes without rewriting them.**
► improve data-hiding and encapsulation.
Question No: 3 (Marks: 1) – Please choose one
Suppose you have been given the following design,
"A person has a name, age, address and ——. You are designing a class to represent a type of person called a patient. This kind of person may be given a diagnosis, have a spouse and may be alive".
Given that the person class has already been created, what of the following would be appropriate to include when you design the patient class?
► **registration date and diagnosis**
► age and ——
► —— and diagnosis
► diagnosis and age
Question No: 4 (Marks: 1) – Please choose one
What problem(s) may occur when we copy objects without using deep copy constructor?
► Dangling pointer
► Memory Leakage
► **All of the given**
► System crash
Question No: 5 (Marks: 1) – Please choose one
this pointers are not accessible for static member functions.
► **True**
► False
Question No: 6 (Marks: 1) – Please choose one
A static member function cannot be declared.
► Static
► **Implicit**

► Explicit
► Virtual
Question No: 7 (Marks: 1) – Please choose one
_____ remain in memory even when all objects of a class have been destroyed.
► Static variables
► Instance variable
► Primitive variables
► None of given
Question No: 8 (Marks: 1) – Please choose one
Friend functions are _____ functions of a class.
► None of given
► object member
► non-member
► data member
Question No: 9 (Marks: 1) – Please choose one
_____, which means if A declares B as its friend it does NOT mean that A can access private data of B. It only means that B can access all data of A.
► Friendship is one way only
► Friendship is two way only
► NO Friendship between classes
► Any kind of friendship
Question No: 10 (Marks: 1) – Please choose one
The statement objA=objB; will cause a compiler error if the objects are of different classes.
► True
► False
The members of a class that can be accessed without creating the object of the class is called

Private member
Data Member
Public Member
Static
==========>

MCQz (Set-18)

==========>

MCQz (Set-20)