

**1. Define object oriented programming?**

OOP is an approach that provides a way of modularizing programs by creating partitioned memory areas for both data and functions that can be used as templates for creating copies of such modules on demand.

**2. List some features of OOP?**

- i. Emphasis is on data rather than procedures.
- ii. Programs that are divided into what are known as objects.
- iii. Follows bottom – up approach in program design.
- iv. Functions that operate on the data of an object are tried together in the data structure.

**3. Define objects?**

An entity that can store data and send and receive messages. It is an instance of a class.

**4. What are the basic concepts of OOPs?**

- i. Objects
- ii. Class
- iii. Data Abstraction
- iv. Encapsulation
- v. Inheritance
- vi. Polymorphism
- vii. Dynamic binding
- viii. Message passing
- ix.

**5. Define class?**

A group of objects that share common properties and relationships. It is a new data type that contains member variables and functions that operate on the variables. A class is defined with a keyword class.

**6. Define data encapsulation?**

A wrapping up of data and functions into a single unit is known as encapsulation.

**7. What is inheritance?**

It is the process by which objects of one class acquire the properties of the another class.

**8. Define operator overloading?**

The process of making an operator to exhibit different behavior in different instances is known as operator overloading.

**9. Define dynamic binding?**

Dynamic binding means that the code associated with a given procedure is not known until the time of the call at run time.

**10. Give some advantages of OOP's?**

i. Through inheritance we can eliminate redundant code and extend the use of existing.

- ii. Classes.
- iii. Easy to partition the work.
- iv. Software complexity can easily be managed.

**11. What are the object oriented languages used?**

- Object based programming languages
- Object oriented programming languages

**12. Give some applications of OOPs?**

- i. Real time systems
- ii. Simulation and modeling
- iii. Object oriented databases
- iv. CIM/CAD systems

**13. Give the structure of a C++ program?**

<b>Include files</b>
<b>Class declaration</b>
<b>Class function definition</b>
<b>Main function program</b>

**14. Write a program to display the following using single cout statement?**

```

Name = XXX
Age = 23
#include <iostream.h>
void main()
{
cout<<"Name = XXX\n"<<"Age = 23";
}

```

**15. What is a comment?**

Comments start with a double slash symbol and terminate at the end of the line. A comment may start any where in the line and whatever follows till the end of the line is ignored.

**16. List some tokens?**

Keywords, identifiers, constants, strings and operators.

**17. Define structures and classes?**

**Structures:-**

A hierarchical set of names that refer to an aggregate of data items may have different attributes.

**Class:-**

A group of objects that share common properties and relationships. A class is defined with a keyword class.

### 18. Define arrays?

A collection of data elements arranged to be indexed in one or more dimensions. The arrays are stored in contiguous memory.

### 19. What are the two ways to create symbolic constant?

- Using a qualifier constant
- Defining a set of integer constants using enum keyword.

### 20. What are the new operators used in C++?

**::** Scope resolution operator  
**::\*** Pointer to member declaration  
**->\*** Pointer to member operator  
**.\*** Pointer to member operator  
**\n** Line feed operator  
**delete** memory release operator  
**setw** Field width operator

### 21. What are the two forms of if statement? Give the syntax also?

The two forms of if statement are

- Simple if statement
- If – else statement

Syntax:-

#### **Form 1:-**

If(expression is true)

```
{  
Action1;  
}
```

Action2;

#### **Form 2:-**

If (expression is true)

```
{  
Action1;  
}  
Else  
{  
Action2;  
}  
Action3;
```

### 22. What is an expression?

An expression is a combination of operators, constants and variables arranged as per rules of the language. It may include function calls which return values.

**23. Define keywords?**

Keywords are explicitly reserved identifiers and cannot be used as names for the program variables or other user defined program elements.

**24. Define pointers?**

A data type that holds the address of a location in memory.

**25. What is an enumerated data type?**

A data type consisting of a named set of values. The C++ compiler assigns an integer to each member of the set.

**26. What is a main function?**

All the C++ programs start with the function main(). Function main returns the integer value that indicates whether the program executed successfully or not.

**Syntax:-**

```
Main()
{
}
```

**27. What is the purpose for the return statement?**

The return statement is used to return the value from a function. The statement return 0; returns the value 0. The return statement supplies a value from the called function to the calling function.

**28. Explain function prototype?**

It is used to describe the function interface to the compiler by giving details such as type number and type arguments and the type of return values. Function prototype is a declaration statement in the calling program.

**Syntax:-**

```
Type function_name (arguments);
```

**29. What is a call by reference?**

A function call mechanism that passes arguments to a function by passing the addresses of the arguments.

**30. Define call by value?**

A function call mechanism that passes arguments to a function by passing a copy of the value of the arguments.

**31. Define macro?**

A short piece of text or text template that can be expanded into a longer text.

**32. What do you inline function?**

A function definition such that each call to the function is in effect replaced by the statements that define the function.

**33. What are the situations that inline functions may not work?**

1. For function returning values, if a loop, a switch, or a goto exists.
2. For function not returning values, if a return statement exists.
3. If function contains static variables.
4. If inline functions are recursive.

**34. What are constant arguments?**

The qualifier const tells the compiler that the function should not modify the argument. The compiler will generate an error when this condition is violated.

**35. Define virtual function?**

A function qualified by the virtual keyword. When a virtual function is called via a pointer, the class of the object pointed to determine which function definition will be used. Virtual functions implement polymorphism whereby objects belonging to different classes can respond to the same message in different ways.

**36. Define friend function?**

A function that has access to the private members of a class but is not itself a member of the class. An entire class can be a friend of another class.

**37. What the two specifications of a class?**

The two specifications of a class are:

Class declaration

Class function definition

**38. What is a class?**

A group of objects that share common properties and relationships. A class is defined with the keyword class.

**39. How is a member function of a class defined?**

Member function of a class is defined in two places:

1. Outside the class definition
2. Inside the class definition

**40. What are the characteristics of a member function?**

- Several different classes can use the same function name. The membership label resolve their scope.
- Member functions can access the private data of the class.
- A member function can call another member function directly, without using the dot operator.

**41. Define nesting of member function?**

A member function can be called by using its name inside another member function of the same class is known as nesting of member functions.

**42. What are the characteristic of a static data member?**

1. It is initialized to zero when the first object of its class is created. No other initialization is permitted.
2. Only one copy of the member is created for the entire class and is shared by all the objects of that class no matter how many objects are created.
3. It is visible only within the class, but its lifetime is the entire program.

**43. Give the properties of the static member functions?**

i. A **static** function can have access to only other static members declared in the same class.

ii. A **static** member function can be called using the class name as class name ::  
function name

**44. How an object as function argument can be done?**

This can be done in two ways:

- a. A copy of the entire object is passed to the function
- b. Only the address of the object is transferred to the function.

The first method is called pass by value and the second method is called pass by reference.

**45. What are the characteristics of the friend function?**

- i. It is not in the scope of the class to which it has been declared as friend.
- ii. It can be invoked like a normal function without the help of any object.
- iii. It has the objects as the arguments.

**46. Define pointer?**

A pointer is a data type that holds the address of a location in memory.

**47. Explain function overloading?**

Function names can be overloaded in C++. We can assign the same name to two or more distinct functions.

(e.g)

```
Int mul(int a, int b);  
Float mul(float x, float y);
```

**48. Define private base classes?**

A base class which allows its public and protected members to be inherited as private members of the private members of the derived class. Thus the inherited members are accessible to the members and friends of the derived class. But they are not accessible the users of the derived class.

**49. Define default argument?**

An argument value that is specified in a function declaration and is used if the corresponding actual argument is omitted when the function is called.

**50. Give the syntax for calling a member function?**

Object name . function name (actual arguments);

**51. Define constructor?**

A special member function automatically creating an instance of a class. This function has the same name as the class.

**52. What are the special characteristics of constructor?**

- They should be declared in the public section
- They are invoked automatically when the objects are created.
- They cannot be inherited.

**53. What is parameterized constructor?**

The constructor that take arguments are called parameterized constructor. When a constructor has been parameterized the object declaration statement such as integer i1 will not work.

**54. What are the two ways in which a parameterized constructor can be called?**

- By calling the constructor explicitly
- By calling the constructor implicitly

**55. What are the kinds of constructors that we call?**

- Constructor without arguments
- Constructor with arguments

**56. What is the advantage of using dynamic initialization?**

The advantage of using dynamic initialization is that various initialization formats can be provided using overloaded constructors.

**57. What are copy constructors? Explain with example?**

The constructor that creates a new class object from an existing object of the same class.

Eg:- integer i2(i1) or integer i2 = i1 would define the object i2 at the same time initialize the values of i1.

**58. Define destructor?**

A function that is called to deallocate the memory of the objects of a class.

**59. List the difference between constructor and destructor?**

Constructor can have parameters. There can be more than one constructor. Constructors is invoked when from object is declared.

Destructor have no parameters. Only one destructor is used in class. Destructor is invoked up on exit program.

**60. Define dynamic constructor?**

Allocation of memory to objects at the time of their construction is known as dynamic constructor.

**61. What is the advantage of using dynamic initialization?**

The advantage of using dynamic initialization is that various initialization formats can be provided using overloaded constructor.

## 62. Define operator overloading?

A language feature that allows a function or operator to be given more than one definition. For instance C++ permits to add two variables of user defined types with the same syntax that is applied to the basic types. The mechanism of giving such special meaning to an operator is known as operator overloading.

## 63. Give the operator in C++ which cannot be overloaded?

- i. Sizeof ->size of operator
- ii. :: ->scope resolution operator
- iii. ?: -> conditional operator
- iv. . ->Membership operator
- v. .\* ->pointer to member operator

## 64. How can we overload a function?

With the help of a special operator called operator function. The general form of an operator function is:

```
Return type class name :: operator op(arg list)
{
Function body
}
```

## 65. Give any four rules for operator overloading?

- (i) Only existing operators can be overloaded.
- (ii) The overloaded operator must have at least one operand that is of user defined type.
- (iii) We cannot use friend functions to overload certain operators.
- (iv) Overloaded operators follow the syntax rules of the original operators.

## 66. What are the steps that involve in the process of overloading?

- Create a class that defines the data type that is to be used in the overloading operation.
- Declare the operator function operator op() in the public part of a class.
- Define the operator function to implement the required operation.

## 67. What are the restrictions and limitations on overloading operators?

Operator function must be member functions or friend functions. The overloading operator must have at least one operand that is of user defined datatype.

## 68. Give a function to overload a unary minus operator using a friend function?

```
Friend void operator -(space &s)
{
s.x = -s.x;
s.y = -s.y;
s.z = -s.z;
}
```



**69. Define unary and binary operator overloading?**

Overloading without explicit arguments to an operator function is known as unary operator overloading and overloading with a single explicit argument is known as binary operator overloading.

**70. Explain overloading of new and delete operators?**

The memory allocation operators new and delete can be overloaded to handle memory resource in a customized way. The main reason for overloading these functions is to increase the efficiency of memory management.

**71. Define type conversion?**

A conversion of value from one data type to another.

**72. What are the three types of conversion takes place?**

- Conversion from basic type to class type
- Conversion from class type to basic type
- Conversion from one class type to another class type.

**73. When and how the conversion function exists?**

To convert the data from a basic type to user defined type. The conversion should be defined in user defined object's class in the form of a constructor. The constructor function takes a single argument of basic data type.

**74. Give the syntax for overloading with friend functions?**

```
Friend return type operator op(arguments)
{
    Body of the function
}
```

**75. Define inheritance?**

The mechanism of deriving a new class from an old one is called inheritance. The old class is referred to as the base class and the new one is called the derived class or the subclass.

**76. What are the types in inheritance?**

- i. Single inheritance
- ii. Multiple inheritance
- iii. Multilevel inheritance
- iv. Hierarchical inheritance
- v. Hybrid inheritance

**77. Explain single inheritance?**

A derived class with only one base class is called single inheritance

**78. What is multiple inheritance?**

A derived class with more than one base class is called multiple inheritance.

**79. Define hierarchical inheritance?**

One class may be inherited by more than one class. This process is known as hierarchical inheritance.

**80. What is hybrid inheritance?**

There could be situations where we need to apply two or more type of inheritance to design a program. This is called hybrid inheritance.

**81. What is multilevel inheritance?**

The mechanism of deriving a class from another derived class is known as multilevel inheritance.

**82. What is virtual base class?**

The child has two direct base classes parent1 and parent2 which themselves have common base class grand parent. The child inherits the traits of grand parent via two separate paths. It can also inherit directly shown by the broken line. The grand parent is sometimes referred to as indirect base class.

**83. What is abstract class?**

An abstract class is one that is not used to create objects. An abstract class is designed only to act as a base class. It is a design concept in program development and provides a base upon which other classes may be built.

**84. What is nesting of classes?**

A class can contain objects of other classes as its members. It is called nesting of classes.

```
Class A{....};  
Class B  
{  
    A a;  
};
```

**85. What is polymorphism?**

Polymorphism is one of the crucial feature of OOP. It simple means one name multiple forms. It contain run time polymorphism and compile time polymorphism.

**86. What are the types of polymorphism?**

- Run time polymorphism
- Compile time polymorphism

**87. What is compile time polymorphism?**

The overloaded member function are selected for invoking by matching arguments both type and number. This information is known to the compiler at the compile time and therefore compiler is able to select the appropriate function for a particular call at the compile time itself. This is called early binding or static binding or static linking. Also known as compile time polymorphism.

**88. What the types in compile time polymorphism?**

- Function overloading
- Operator overloading

**89. What is this pointer?**

A unique keyword called this to represent an object has invokes a member function.

**90. What is virtual function?**

The same function same in both the base and derived classes the function in base class is declared as virtual using the keyword virtual preceding its normal function declaration.

**91. What are the rules for virtual function?**

1. They cannot be static members
2. They are access by using object pointers
3. A virtual function can be a friend of another class.

**92. What is pure virtual function?**

A pure virtual function is a function declared in a base class that has no definition relative to the base class.

**93. What are the file stream classes in C++?**

1. filebuf
2. fstreambase
3. ifstream
4. ofstream
5. fstream

**94. What are the file manipulation function in C++?**

- i. seekg()
- ii. seekp()
- iii. tellg()
- iv. tellp()

**95. What are the file open modes?**

ios::app, ios::binary, ios::out, ios::in, ios::ate, ios::nocreate, ios::noreplace, ios::trunk.

**96. What are the error handling function in C++?**

1. eof()
2. fail()
3. bad()
4. good()

## 16 Marks

### 1. Describe principles of object oriented programming?

#### a. Classes

A group of objects that share common properties and relationships. It is a new data type that contains member variables and functions that operate on the variables. A class is defined with a keyword class.

#### b. Objects

#### c. Abstraction

#### d. Encapsulation

A wrapping up of data and functions into a single unit is known as encapsulation.

#### e. Inheritance

It is the process by which objects of one class acquire the properties of the another class.

#### f. Polymorphism

#### g. Dynamic binding

Dynamic binding means that the code associated with a given procedure is not known until the time of the call at run time.

#### h. Message passing

### 2. Explain object oriented languages?

#### a. Object based programming languages

#### b. Object oriented programming languages

### 3. What are the applications of OOPs?

Advantages:-

i. Through inheritance we can eliminate redundant code and extend the use of existing.

ii. Classes.

iii. Easy to partition the work.

iv. Software complexity can easily be managed.

**Applications:-**

Real time systems

ii. Simulation and modeling

iii. Object oriented databases

iv. CIM/CAD systems

### 4. Explain inheritance concept?

➤ **Single inheritance**

➤ Multiple inheritance

➤ Multilevel inheritance

➤ Hierarchical inheritance

➤ Hybrid inheritance

**5. Explain polymorphism?**

Compile time polymorphism

Run time polymorphism