

MARIA COLLEGE OF ENGINEERING & TECHNOLOGY,

ATTOOR

DEPARTMENT OF CSE & IT

TWO MARKS QUESTIONS WITH ANSWERS

PREPARED BY

VIJIS.L

LECTURER

UNIT I & II

1. Define object oriented programming.

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

2. List out the features of OOPS.

- Classes
- Objects
- Data abstraction
- Data encapsulation
- Inheritance
- Polymorphism
- Message passing
- Extensibility
- Persistence
- Delegation
- Generosity
- Dynamic binding.

3. What is an object?

Objects are the basic run time entities in an object oriented system. They may represent a person, a place or any data item. Objects contain data and code to manipulate data .

4. What is a class?

The entire set of data and code of an object that can be made a user defined data type with the help of a class. A class is a collection of objects of type class.

5. What is data abstraction ?

The technique of creating new data types that are well suited to an application to be programmed is known as data abstraction.

6.What is data encapsulation?

The wrapping up of data and functions into a single unit called class is known as data encapsulation.

7.What is data hiding ?

The insulation of the data from direct access by the program is called data hiding or information hiding.

8. What is inheritance / reusability / derivation ?

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

The concept of inheritance provides the idea of reusability. This means that we can add additional features to an existing class without modifying it .

Derivation involves the creation of new classes (derived class) from the existing ones (base class).

9.What is polymorphism ?

Polymorphism means the ability to take more than one form.

10. What is extensibility ?

It is a feature which allows the extension of the functionality of the existing software components. In C++ this is achieved through abstract classes and inheritance.

11.What is persistence ?

The phenomenon where the object outlives the program execution time and exists between the executions of a program is known as persistence.

12. What is delegation ?

It is an alternative to class inheritance. In delegation two objects are involved in handling a request. This is analogous to the child classes sending requests to the parent classes.

13. What is generosity?

It is a technique for defining software components that have more than one interpretation depending on the data type of parameters. The unknown data types are resolved at the time of their usage based on the data type of parameters.

14. What is dynamic binding?

Binding refers to the linking of a procedure call to the code to be executed in response to the call. Dynamic binding also known as late binding means that the code associated with a given procedure call is not known until the time of call at run time.

15 . What is the use of a break statement ?

A break construct terminates the execution of loop and the control is transferred to the statement immediately following the loop. The term break refers to the act of breaking out of a block of code.

16. What is the use of a continue statement ?

The continue statement skips the remainder of the current iteration initiates the execution of the next iteration.

17. List out the string functions ?

- String length
- String copy
- String concatenation
- String comparison
- String conversions

18. Define function.

The process of splitting a large program into small manageable tasks and designing them independently is popularly called modular programming or divides and conquers technique. A repeated group of instructions in a program can be organized as a function. A function is a set of program statements that can be processed independently.

19. What are the advantages of using functions ?

- Modular programming
- Reduction in the amount of work and development time
- Program and function debugging is easier.
- Division of work is simplified.
- Reduction in the size of the program.

20. What are the components of a function?

- Function declaration or function prototype.
- Function parameters and formal parameters.
- Function definition
- Return statement
- Function call

21. What are the types of parameter passing mechanism ?

- Pass by value
- Pass by address
- Pass by reference.

22. What is an inline function?

Inline functions are those whose function body is inserted in place of the function call statement during the compilation process. With the inline code the program will not incur any context switching overhead.

23. What is a recursive function?

A function that contains a function call to itself or a function call to a second function which eventually calls the first function is known as recursive functions.

24. What is a storage class? What are types of storage classes?

The extent of the variable is characterized by storage classes.

Types:

- Auto
- Register
- Extern
- Static

25. What is an auto variable?

By default, all the variables are defined as auto variables. They are created when the function or block is entered and destroyed when the function or block is terminated.

26. What is a register variable?

The allocation of CPU registers to variables speeds up the execution of a program; memory is not referred when such variables are accessed.

27. What is a static variable?

The static storage class allows to define a variable whose scope is restricted to either a block, a function, or a file and extent is the life span of the program.

28. What is an extern variable?

Global variables must be defined only once in any of the program module and they can be accessed by all others. It is achieved by declaring such variables as extern variables.

29. What is data conversion?

When we use the assignment operator we assign a value on the right hand side to a variable on the left side & if it is of a different data type then we have to perform data conversion or type conversion.

30. What are the methods of data conversion?

There are two methods for data conversion:

- (i) Implicit data conversion
- (ii) Explicit data conversion

31. What is meant by explicit data conversion?

Sometimes we want to force the compiler to convert one data type to another. To do this we use the cast operator. For example, to convert float into int we should say `intvar= int(floatvar);`

32. What is type casting?

Casting provides explicit conversion between type for example if we want to assign a float variable to an integer we have to use:

```
intvar=int (floatvar)
```

33. What are the operators which cannot be overloaded?

The following operators cannot be overloaded:

- (i) the member access or data operator (.)
- (ii) the scope resolution operator (: :)
- (iii) the conditional operator (? :)
- (iv) pointer to member operator(. *)

34. Give the structure of a C++ program.

Include files

Class definition

Member function definitions

Main function

35.What are the parts of a function ?

The parts of a function are :

FUNCTION HEADER

- Function type / return type
- Function name
- Parameter / argument list

FUNCTION BODY

- Variable declaration
- Executable statements
- Return statement

36.What is function prototype ?

Function prototype is otherwise known as function declaration. The prototype describes the function interface to the compiler by giving details such as the number and type of arguments and the type of return values.

37.What is a function call ?

When the calling function invokes the called function then it is known as function call.

38.What do you mean by function definition ?

The implementation of requirement or a set of instructions used to specify a task is known as function definition.

39.What is an inline function ?

An inline function is a function that is expanded in line when it is invoked. The inline functions are defined as follows :

Inline function-header

{

Function body

}

ex : inline double cube(double a)

```
{  
    Return(a*a*a);  
}
```

40.write about the situations where inline expansion may not work ?

- for functions returning values, if a loop, a switch, or a goto exists.
- for functions not returning values, if a return statement exists.
- If functions contain static variables.
- If inline functions are recursive.

41.What is a class ? How will you define a class ?

A class is a way to bind the data and its associated functions together. A class specification has two parts :

- Class declaration
- Class function definition

42.Give the general form of a class declaration ?

The class declaration describes the type and scope of its members. The class function definitions describe how the class functions are implemented.

```
Class classname  
  
{  
    private :  
        variable declarations;  
        function declarations;
```

```
public :  
  
    variable declarations;  
  
    function declarations;  
  
};
```

43.What are visibility labels ?

The keywords private , protected and public are known as visibility labels or access control specifies. The members that have been declared as private can be accessed only from within the class. Protected members can be accessed by base class and derived class. On the other hand, public members can be accessed from outside the class also.

44.What are the characteristics of a member function ?

- Several different classes can use the same function name. The membership label will resolve their scope.
- Member functions can access the private data of the class. A non member function cannot do so.
- A member function can call another member function directly, without using the dot operator.

45.What are the characteristics of a static data member ?

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

46.What are the properties of a static member function ?

- A static function can have access to only other static members, declared in the same class.
- A static member function can be called using the class name as follows

```
Classname :: function-name;
```

47. In what way is a private member function different from public member function.

A private member function can only be called by another function that is a member of its class. Even an object cannot invoke a private function using the dot operator.

48. Give the syntax to define member function outside the class definition ?

```
Return-type class-name :: function-name ( argument declaration )  
  
{  
  
    function body ;  
  
}
```

49. What is a constructor ?

A constructor is a special member function whose task is to initialize the objects of its class. It is special because its name is the same as the class name. The constructor is invoked whenever an object of its associated class is created. It is called constructor because it constructs the values of data members of the class.

50. How will you declare and define a constructor ?

Constructor declaration :

```
Class integer  
  
{  
  
    int m,n;  
  
    public :  
  
        integer ( void );  
  
}
```

constructor definition :

```
integer :: integer (void )  
  
{
```

```
m=0;
n=0;
}
```

51.what are the characteristics of a constructor ?

- They should be declared in the public section.
- They are invoked automatically when the objects are created.
- They do not have return types, not even void and therefore, they cannot return values.
- They cannot be inherited, though a derived class can call the base class constructor.
- Constructors cannot be virtual.

52.what are the types of a constructor ?

- Default constructor
- Parameterized constructor.
- Copy constructor

53.What is a parameteized constructor ?

The constructors that can take arguments are called parameterized constructors.

```
Integer ( int x, int y )
{
    m=x;
    n=y;
}
```

54.What is a default constructor ?

A constructor that accepts no parameter is called default constructor.

```
Default ( )
{
}
```

55. Give an example for multiple constructors in a class ?

```
Class integer
{
    int m,n;
    public :
        integer ( )
        {
            m=0;
            n=0;
        }
        integer ( int a, int b)
        {
            m=a;
            n=b;
        }
        integer ( integer & i )
        {
            m=i.m;
            n=i.n;
        }
};
```

56. What is a destructor ?

A destructor as the name implies is used to destroy the objects that have been created by a constructor. Like a constructor the destructor is a member function whose name is the same as the class name but is preceded by a tilde symbol.

```
~ Integer ( )  
  
 {  
  
 }
```

57. Define operator overloading ?

The mechanism of giving special meaning to an operator is known as operator overloading. We can overload all the operators except the following.

- Class member access operator
- Scope resolution operator
- Sizeof operator
- Conditional operator

58. Give the general form of an operator function ?

```
Return_type class_name :: operator op ( argument _ list )
```

```
 {  
  
     Function body ;  
  
 }
```

59. Write the steps involved in operator overloading ?

- First , create a class that defines the data type that is to be used in the overloading operation
- Declare the operator function in the public part of a class. It may be either a member function or a friend function.
- Define the operator function to implement the required operations.

60. Define dynamic memory allocation ?

The technique of allocating memory during run time on demand is known as dynamic memory allocation.

61.What are the drawbacks of static memory allocation ?

- Excess wastage of memory or insufficient memory .
- Inefficient , because memory allocated is not freed .

62.What is the use of delete operator ?

Delete operator is used to return the memory allocated by the new operator back to the memory pool. Memory thus released will be reused by other parts of the program .

63.What is a pointer ? What are the uses of a pointer ?

Pointer is defined as a variable used to store memory addresses.

Uses :

- Accessing array elements .
- Passing arguments to a function when the function needs to modify the original.
- Passing arrays and strings to functions.
- Creating data structures such as linked lists, binary tree etc .
- Obtaining memory from the system dynamically.

64.What is a friend function ?

Friend function is a special type of function which is used to access all the private and protected members of a class. The functions that are declared with the keyword friend are called friend functions. A function can be a friend to multiple classes.

65.What are the properties of a friend function ?

- A friend function is not in the scope of the class to which it has been declared as friend.
- It can be invoked like a normal function without the help of any object.
- Unlike member functions it cannot access the member names directly and has to use an object name and dot membership with each member name.

66. What is the difference between friend function and member function ?

The only difference between a friend function and member function is that, the friend function requires the argument to be explicitly passed to the function and processes them explicitly, whereas, the member function considers the first argument implicitly.

67. What is default arguments.

Default arguments

C++ allows us to call a function without specifying all its arguments. In such cases the function assigns a default value to the parameter which does not have a matching argument in the function call.

Ex: function prototype

```
Float amount(float a,int b,int c=100);
```

Function call

```
Val=amount(10,20);
```

68. What is Static class member

A data member of class can be declared as static.

Properties:

1. It is initialized to 0 when the first object of its class is created.
2. No other initialization permitted.
3. Only one copy of that member is created for the entire class.
4. It is visible only within the class.

69. What is Static member function

Like static member variable we can also have static member functions.

Properties:

1. A static function can have access only other static members (functions or variables) declared in the same class.
2. A static member function can be called using the class name instead of its objects.

Class name:: function_name

70. what is Dynamic constructors

The constructors can also be used to allocate memory while creating objects. This will enable the system to allocate the right amount of memory for each object when the objects are not of the same size thus resulting in the saving of memory. Allocation of memory to objects at the time of their construction is known as dynamic constructors. The memory is allocated with the help of the new operator.

71.What is function overloading? Give an example.

Function overloading means we can use the same function name to create functions that perform a variety of different tasks.

Eg: An overloaded add () function handles different data types as shown below.

// Declarations

i. int add(int a, int b); //add function with 2 arguments of same type

ii. int add(int a, int b, int c); //add function with 3 arguments of same type

iii. double add(int p, double q); //add function with 2 arguments of different type

//Function calls

add (3 , 4); //uses prototype (i.)

add (3, 4, 5); //uses prototype (ii.)

add (3 , 10.0); //uses prototype (iii.)

72) What is operator overloading?

C++ has the ability to provide the operators with a special meaning for a data type.

This mechanism of giving such special meanings to an operator is known as Operator overloading. It provides a flexible option for the creation of new definitions for C++ operators.

73) List out the operators that cannot be overloaded.

_ Class member access operator (. , .*)

_ Scope resolution operator (::)

_ Size operator (sizeof)

_ Conditional operator (?:)

74) What is the purpose of using operator function? Write its syntax.

To define an additional task to an operator, we must specify what it means in relation to the class to which the operator is applied. This is done by Operator function , which describes the task. Operator functions are either member functions or friend functions. The general form is

```
return type classname :: operator (op-arglist )
```

```
{
```

```
function body
```

```
}
```

where *return type* is the type of value returned by specified operation.

Op-operator being overloaded. The *op* is preceded by a keyword operator. operator *op* is the function name.

75) Write at least four rules for Operator overloading.

- _ Only the existing operators can be overloaded.
- _ The overloaded operator must have at least one operand that is of user defined data type.
- _ The basic meaning of the operator should not be changed.
- _ Overloaded operators follow the syntax rules of the original operators.

They cannot be overridden.

76) How will you overload Unary & Binary operator using member functions?

When unary operators are overloaded using member functions it takes no explicit arguments and return no explicit values.

When binary operators are overloaded using member functions, it takes one

explicit argument. Also the left hand side operand must be an object of the relevant class.

77) How will you overload Unary and Binary operator using Friend functions?

When unary operators are overloaded using friend function, it takes one reference argument (object of the relevant class)

When binary operators are overloaded using friend function, it takes two explicit arguments.

78) How an overloaded operator can be invoked using member functions?

In case of Unary operators, overloaded operator can be invoked as

op object_name or object_name op

In case of binary operators, it would be invoked as

Object . operator op(y)

where op is the overloaded operator and y is the argument.

79) How an overloaded operator can be invoked using Friend functions?

In case of unary operators, overloaded operator can be invoked as

Operator op (x);

In case of binary operators, overloaded operator can be invoked as

Operator op (x , y)

80) List out the operators that cannot be overloaded using Friend function.

_ Assignment operator =

_ Function call operator ()

_ Subscripting operator []

→_ Class member access operator

81) What is meant by casting operator and write the general form of overloaded casting operator.

A casting operator is a function that satisfies the following conditions

- _ It must be a class member.
- _ It must not specify a return type.
- _ It must not have any arguments.

The general form of overloaded casting operator is

```
operator type name ( )  
  
{  
  
..... // function statements  
  
}
```

It is also known as conversion function.

82. Explain basic to class type conversion with an example.

Conversion from basic data type to class type can be done in destination class.

Using constructors does it. Constructor takes a single argument whose type is to be converted.

Eg: Converting int type to class type

```
class time  
  
{  
  
int hrs,mins;  
  
public:  
  
.....  
  
Time ( int t)//constructor
```

```

{
hours= t/60 ; //t in minutes
mins =t % 60;
}
};

```

Constructor will be called automatically while creating objects so that this conversion is done automatically.

83) Explain class to basic type conversion with an example.

Using Type Casting operator, conversion from class to basic type conversion can be done. It is done in the source class itself.

Eg: `vector : : operator double()`

```

{
double sum=0;
for(int I=0;I<size;I++)
sum=sum+v[ i ] *u[ i ] ;
return sqrt ( sum ) ;
}

```

This function converts a vector to the corresponding scalar magnitude.

84) Explain one class to another class conversion with an example.

Conversion from one class type to another is the combination of class to basic and basic to class type conversion. Here constructor is used in destination class and casting operator function is used in source class.

Eg: `objX = objY`

Unit III

1. What is an exception ?

Exception refers to unexpected condition in a program. The unusual conditions could be faults, causing an error which in turn causes the program to fail. The error handling mechanism of C++ is generally referred to as exception handling.

2. What are the types of exception ?

They are classified into synchronous and asynchronous exceptions.

Synchronous exception :

The exception which occur during program execution , due to some fault in the input data or technique that is not suitable to handle the current class of data, within the program are known as synchronous exception. For instance errors such as out-of-range, overflow, underflow and so on.

Asynchronous exception :

The exceptions caused by events or faults unrelated to the program and beyond the control of the program are called asynchronous exception. For instance, errors such as keyboard interrupts, hardware malfunctions, disk failure, and so on.

3. What are the blocks related to exception handling constructs ?

The blocks related to exception handling constructs are

- try
- throw
- catch

The keyword try is used to preface a block of statements. Which may generate exceptions. This block of statements is known as try block.

When an exception is detected, it is thrown using throw statement in the try block.

A catch block catches the exception thrown by the throw statement in the try block and handles it appropriately.

Syntax:

```
try
```

```
{
```

.....

.....

throw exception;

}

catch (type arg)

{ }

4. What are the functions supported by C++ to handle uncaught exceptions ?

The functions supported by C++ to handle uncaught exceptions are

terminate ()

set_terminate ()

unexpected ()

set_unexpected ()

5. What is a template ?

Templates support generic programming , which allows to develop reusable software components such as functions, classes etc., supporting different data types in a single framework.

6. What is function template ?

The templates declared for functions are called function templates. They perform appropriate operations depending on the data type of the parameters passed to them.

7. What is a class template ?

The templates declared for classes are called function templates. They perform appropriate operations depending on the data type of the parameters passed to them.

8. Define runtime exceptions

Runtime exceptions are automatically defined for the programs that we write and include things such as division by zero and invalid array indexing.

9. Give the general form of throw .

`throw throwableinstance;`

throwable instance must be an object of type throwable or a subclass of throwable.

UNIT IV & V

1. Define inheritance ?

Inheritance is the process of creating new classes from the existing classes. The new classes are called derived classes. The existing classes are called base classes. The derived classes inherit all the properties of the base class plus its own properties. The properties of inheritance does not affect the base classes.

2. What are the types of inheritance ?

- **Single inheritance**
- **Multiple inheritance**
- **Hierarchial inheritance**
- **Multilevel inheritance.**
- **Hybrid inheritance**
- **Multipath inheritance**

3. What are the advantages of using a derived class ?

- **New classes can be derived by the user from the existing classes without any modification.**
- **It saves time and money.**
- **It reduces program coding time.**
- **It increases the reliability of the program.**

4. Define single inheritance ?

If a derived class is derived from a base class then it is called single inheritance.

5. Define multiple inheritance ?

If a derived class is derived from more than one base class, then it is called multiple inheritance.

6. Define hierarchial inheritance ?

If two or more derived classes are derived from the same base class then it is known as hierarchial inheritance.

7. Define multilevel inheritance ?

If a derived class is derived from another derived class then it is known as multilevel inheritance.

8. Define hybrid inheritance ?

The combination of one or more types of inheritance is called hybrid inheritance.

9. Define multipath inheritance ?

When a class is derived from two or more classes which are derived from the same base class then such type of inheritance is called multipath inheritance.

10. Define abstract class ?

A class is said to be an abstract class if it satisfies the following conditions

- It should act as a base class**
- It should not be used to create any objects**

10b. What is an virtual function ?

A member function whose definition can be changed during run time is called virtual function. The class which contains virtual function is called polymorphic class and it should be a base class. Different versions for the virtual function should be present in different derived classes with same name as virtual function name.

11. Define pure virtual function ?

pure virtual function is a virtual function with no body. The general form is :

Virtual void member-function-name() = 0;

12. What are the properties of pure virtual function ?

- **It has no definition in the base class.**
- **We cannot create object for the class containing pure virtual function.**
- **Pure virtual function can be called through derived class objects.**
- **It acts as an empty bucket which has to be filled by its derived classes.**

13. What are the rules for virtual functions ?

- **Virtual functions must be members of some class.**
- **They cannot be static members.**
- **They are accessed by using object pointers.**
- **A virtual function can be a friend of another class.**
- **We can have virtual destructors, but we cannot have virtual constructors.**
- **A virtual function in a base class must be defined even though it may not be used.**
- **A pointer to a derived class cannot point an object of base type.**
- **When a pointer points to a derived class, incrementing or decrementing will not make it point to the next object of the derived class.**

14. Define Polymorphism

It is the property of the same object to behave differently in different contexts given the same message. A single function name can be used for various purposes and single operator is used to achieve multiple operations and the usage of either the function or the operator depends on the context in such cases.

15. Define Compile time polymorphism

The compiler while compiling the program resolves the function call or the operator call. This is known as compile time polymorphism

16. Define Runtime polymorphism

During multiple inheritance if the appropriate member function could be selected while the program is running is known as Runtime polymorphism

17. Define Run time type information:

It is a very powerful tool in c++ for finding out the type of an object at runtime. Because of runtime functioning, RTTI also impacts the performance of the system to a notable extent.

18.what is a stream ?

Stream is a series of bytes, which act either as a source from which input data can be extracted or as a destination to which the output can be sent. The source stream provides data to the program called to input stream and the destination stream that receives data from the program is called the output stream.

19.what are the types of standard streams ?

- **cin - Standard input corresponding to stdin in C**
- **cout – Standard output corresponding to stdout in C**
- **cerr – Standard error output corresponding to Stderr in C**
- **clog – A fully buffered version of cerr.**

20.How do you classify ios class ?

Istream – input stream does formatted input.

Ostream – output stream does formatted output.

Iostream – input / output stream does formatted input and output.

21.What are manipulators ?

Manipulators are special functions that are specifically designed to modify the working of a stream. They can be embedded in the I/O statements to modify the form parameters of a stream.

22. Give few ios class functions and flags ?

Function	Task performed
Width()	Specifies the required number of fields to be used while displaying the output value.
Precision ()	Specifies the number of digits to be displayed after the decimal point.
Fill ()	Specifies a character to be used to fill the unused area of a field. By default, fills blank space character.
Setf ()	Sets format flag that control the form of output display.
Unsetf ()	Clears the specified format flag.

23. Storing programs and data permanently in main memory is not preferred . Give reasons .

- **Main memory is usually too small to permanently store all the needed programs and data.**
- **Main memory is a volatile storage device, which loses its contents when power is turned off.**

24 What is a file ?

A file is a collection of related information defined by its creator. Commonly files represent programs (both source and object forms) and data. Files may be free-form, such as text files or may be rigidly formatted . In general, a file is a sequence of bits, bytes, lines, or records whose meaning is defined by its creator and user.

25 How many classes are used for handling files ?

ifstream – for handling input files.

ofstream – for handling output files.

fstream – for handling files on which both input and output can be performed.

26. What are the steps involved in manipulating a file ?

- **Name the file on the disk**
- **Open the file to get the file pointer.**
- **Process the file. (Read / Write)**
- **Check for errors while processing.**
- **Close the file after its complete usage.**

27. What functions are used for manipulation of file pointers ?

- **seekg () – Moves get file pointer to a specific location.**
- **seekp () - Moves put file pointer to a specific location.**
- **tellg () – Returns the current position of the get pointer**
- **tellp () - Returns the current position of the put pointer**

28. What do you mean by sequential access ?

A sequential file has to be accessed sequentially ; to access the particular data in the file all the preceding data items have to be read and discarded. For example a file on a tape must be accessed sequentially.

29. What do you mean by random access?

A random file allows access to the specific data without the need for accessing its preceding data items. However, it can be accessed sequentially. For example, a file on a hard disk or floppy disk can be accessed either sequentially or randomly.