GOVERNMENT ARTS AND SCIENCE COLLEGE PERAVURANI-614804

16SCCCS2- PROGRAMMING IN C++

Two Marks Questions & Answers

DEPARTMENT OF COMPUTER SCIENCE

Class: I B.Sc., Computer Science

COMPLIED BY,

Mrs. S.Jamuna M.Sc., M.Phil, B.Ed.,
Guest Lecturer, Dept. of Comp. Sci.
GASC, Peravurani-614804.

Unit- I to V

Two marks Questions & Answers

1. What are the Concepts of OOPs

- Objects
- Classes
- Data Abstraction
- Data Encapsulation
- Inheritance
- Polymorphism
- Message Passing
- Dynamic Binding

2. Define object oriented programming.

Object oriented Programming is defined as 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 copies of such modules on demand. Writing object-oriented programs involves creating classes, creating objects from those classes, and creating applications, which are stand-alone executable programs that use those objects.

3. What is C++ Programming Language?

C++ is objected oriented programming language, It is used to develop the application and system program. C++ was invented in 1979 by Bjarne Stroustrup at Bell Laboratories in New Jersey.

4. What is Data Abstraction?

Data Abstraction is defined as a collection of Data and functions. Since the classes use the concept of data abstraction. They are known as Abstract Data Types(ADT). The only way to access the data is provided by the functions, which are wrapped in the class. Data is not accessible to the outside world.

5. Differentiate Procedure Oriented Programming(POP) and Object Oriented Programming (OOP)

POP	OOP
Emphasis on non-real time	Emphasis on real item
Programs are divided into functions	Programs are divided into Objects
Data are sharable	Data are not sharable
Structured Programming	Object Oriented Programming
Top-Down Approach	Bottom-Up Approach

6. Define Tokens

Smallest individual unit in a program. C++ tokens are Keywords, Identifiers, Constants, Strings, Operators.

7. What are the Data Types in C++?

- Built-in Data types
- User Defined Data types
- Derived Data Types

8. Write the Block Structure of C++

- Include Files
- Class Declaration
- Member Function Definitions
- Main Function Program

9. What are the Operators in C++

- Scope Resolution Operator : :
- Pointer-to-Pointer Member Declarator : :*
- Pointer-to-Pointer Member Operator ->*
- Pointer-to-Pointer Member Operator .*
- Delete-Memory Release Operator

- Endl-Line feed operator
- New-Memory allocation operator
- Setw-Memory width operator

10. List out the expressions in C++?

- Constant Expressions
- Integral Expressions
- Float Expressions
- Pointer Expressions
- Relational Expressions
- Logical Expressions
- Bitwise Expressions

11. What is meant by Data Hiding?

Data are hidden inside a class, that can not be accessed by any function outside the class. It is achieved by declaring the data part as private.

12. What is Polymorphism and its types of Polymorphism?

Polymorphism means many forms. Types of Polymorphism are

- Runtime Polymorphism
- Compile time Polymorphism

13. What are the Application of oops Programming?

- The Areas for application of oop includes
- Real-time system.
- Simulation and modeling
- Object-oriented data base.
- Hypertext ,hypermedia
- AI and expert system
- Neural networks and parallel programming
- Decision support and office automation system.
- CIM/CAD/CAM. System.

14. What are the Components of a Class?

- A Class consists of two components
- Data Members
- Methods or Member Functions

15. What are the Access Specifies Used in a Class?

The Access Specifies

- Public Specifier
- Private Specifier
- Protected Specifier

16. What is Data Hiding?

The private data members cannot be accessed outside of class only access within the class. This process is call data hiding.

17. What do you mean by Private member?

A member declared as Private is a Private member. We cannot access private members anywhere in the program. Private members can be accessed only through Member function of that class. Example : private: int k1,k2;

18 .Define Class.

It is a representation (or) blueprint of an object.

- The entire set of data and code of an object can made a user defined data type with the help of a class.
- Classes are user-defined data types and behave like built-in data types of Programming language and they are known as Abstract Data Type.
- For Example: If Fruit has been declared as a class, then the statement Fruit mango; will create an object **mango** belonging to the class **Fruit.**

19. Define Objects.

- Objects are the basic run-time entities in an object oriented programming.
- It is an instance of class.
- Each instance of an object can hold its own relevant data.
- They may represent a person, a place, a bank account or any item that the program has to handle.
- Each object contains data (data members) and code (member functions) to manipulate the data.

Object: STUDENT
DATA: Name Roll No Mark 1, Mark 2,
FUNCTIONS Total Average Display

20. Define Constructor.

It is a member function having name of its class. It is executed automatically when object is created. It is used to initialize object and allocate the necessary memory.

21. Define Destructor.

It is a member function having the char ~ followed by name of its class. It is executed automatically when object goes out of scope. A class must have only one constructor.

22. Define Constructor Overloading.

A class can have multiple constructors. This is called constructor overloading.

23. What is order of Constructor and Destructor

When more than one object is created, they are destroyed in the reverse Chronological order. Object created must recently is the first one to be destroyed.

24. What is meant by Parameterized constructors.

Constructor that can take arguments are called parameterized constructor.

25. What is meant by Copy Constructors?

It is used to declare and initialize an object from another object

For example , Integer i2 (i1)

Define I2 and at the same time initialize it to the values of i1.

26. Define dynamic constructor

Allocation of memory to objects at time of their construction is known as dynamic constructor. The memory is allocated with the help of the **NEW operator**

Eg:

```
Class string
{
char *name;
int length;
public:
string( )
{
length=0;
name=new char[length +1];
}
void main( )
{
string name1("Louis"),name3(Lagrange);
}
```

27. Define const object

We can create constant object by using const keyword before object declaration.

Eg: Const matrix x(m,n);

28. Write some special characteristics of 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, and they cannot return values
- They cannot be inherited, though a derived class can call the base class
- They can have default arguments
- Constructors cannot be virtual function

29. How the objects are initialized dynamically?

To call parameterized constructor we should the pass values to the object ie,for the constructor integer(int a,int b),it is invoked by integer a(10,18) this value can be get during run time. i.e., for above constructor

```
int p,q;
cin>>p>>q;
integer a(p,q);
```

30. Define Inline Function?

Inline function is defined as a function definition such that each call to the function is in effect, replaced by the statements that define the function. It is expanded in line when it is invoked. The general form is

```
inline function-header
{
function body
}
```

31. Explain return by reference with an example.

A function can also return a reference. Consider the following function

```
int & max( int &x , int &y)
{
if(x>y)
return x;
```

else
return y;
}

Since the return type of max ( ) is int & the function returns reference to x or y. Then a function call such as
max (a , b) will yield a reference to either a or b depending on their values. The statement max (a , b) = -1; is legal and assigns –1 to a if it is larger, otherwise –1 to b.

32. What are Friend functions? Write the syntax

A function that has access to the private member of the class but is not itself a member of the class is called friend functions.

The general form is : friend data_type function_name( );

Friend function is preceded by the keyword ‘friend’.

33. Write some properties of friend functions.

Friend function is not in the scope of the class to which it has been declared as friend. Hence it cannot be called using the object of that class.Usually it has object as arguments. It can be declared either in the public or private part of a class. It cannot access member names directly. It has to use an object name and dot membership operator with each member name. eg: (A . x).

34. What are virtual functions?

A function qualified by the ‘virtual’ keyword is called virtual function. When a virtual function is called through a pointer, class of the object pointed to determine which function definition will be used.

35. Write some of the basic rules for virtual functions

- Virtual functions must be member of some class. They cannot be static members and they are accessed by using object pointers
- Virtual function in a base class must be defined. Prototypes of base class version of a virtual function and all the derived class versions must be identical.
- If a virtual function is defined in the base class, it need not be redefined in the derived class.

36. What are pure virtual functions? Write the syntax.

A pure virtual function is a function declared in a base class that has no definition relative to the base class. In such cases, the compiler requires each derived class to either define the function or redeclare it as a pure virtual function. A class containing pure virtual functions cannot be used to declare any object of its own. It is also known as "donothing" function.

The "do-nothing" function is defined as follows:

```
virtual void display ( ) =0;
```

37. Define Friend Function.

Private members cannot be accessed from outside the class. To make an outside function "Friendly" to a class, declare this function as a friend of the class.

38. What is meant by Friend Class?

We can also declare all the member function of one class the friend of another class. In such cases , the class is called a friend class.

39. What are the Special Characteristics of Friend Function?

- The function definition does not use friend keyword
- It is not in the scope of the class which is declared as friend
- It can be called like normal function without the help of any object
- Friend function acts as a Bridge between 2 classes

40. Define Operator Overloading?

Mechanism of giving such special meanings to an operator is known as Operator Overloading.

41. What are the Operators of C++ that cannot be overloaded?

- . , .* - class member access operator
- :: - Scope Resolution Operaotr
- Sizeof-Size of Operator
- ?:- Conditional Operator

42. Define Inheritance.

Creating new class from old class. (or) Deriving a new class from old class.

43. What are types of Inheritance?

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hybrid Inheritance
- Hierarchical Inheritance

44. What are visibility modes of Inheritance?

Private

Public

Protected

Note: Private members are not inheritable, inaccessible to the objects of derived class.

45. How can you define member functions in c++?

- Defined inside the class
- Defined outside the class

46. What is meant by Abstract Class?

It is the one that is not used to create objects. That is, abstract class is designed only to act as a base class.

47. What is meant by intermediate base class?

In multilevel inheritance, first level derived class is known as intermediate base class.

48. What is meant by Automatic Initialization of objects.

C++ provides a special member function called the constructor which enables an object to initialize itself when it is created.

49. What is meant by Hybrid Inheritance?

Two or more types of inheritance used to derive a class. 2 or set of class acts as a base class, from which we can derive a new class.

50. What is meant by Multipath Inheritance?

It is Consists of multiple, multilevel and hierarchical inheritance.

51. Define Virtual Base Class.

Duplication of inherited members due to multiple paths can be avoided by making the common base class as virtual base class.

52. Define Virtual Function?

It is used to invoke exact version of the member function. Virtual functions should be defined in the public section of a class.

53. How can you access the virtual functions.

Virtual functions have to be accessed through a pointer to the base class. It is not accessible directly.

54. What are the types of type conversion?

- conversion from basic type to class type
- conversion from class type to basic type
- conversion from one class type to another

55. Define operator overloading?

The mechanism of giving such special meanings to an operator is known as operator overloading. or In c++ you can give special meanings to operators when they are used with user defined classes. This is called operator overloading.

56. Why is it necessary to overload an operator?

To define a new relation task to an operator, we must specify what it means in relation to the class to which the operator is applied. This is done with the help of a special function called operator function.

(Or) It allows the developer to program using notation closer to the target domain and allow user types to look like types built into the language. (Or) The ability to tell the compiler how to perform a certain operation when its corresponding operator is used on one or more variables.

57. What is a conversion function? How it is created? Explain its syntax

The type of data to the right of an assignment operator is automatically converted to the type of the variable on the left. For e.g., the statements

```
int m;
float x=3.14;
m=x;
```

Convert x to an integer before its value is assigned t0 m. thus the fractional part is truncated.

58. When is a friend function compulsory? Give an eg.

A friend function is necessary when you an function outside the class. And to access the private members of the class or the member function and also friend class can directly access the private and protected data.

59. 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.)

60. 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.

61. 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.

62. 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.

63. 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).

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

- Assignment operator =
- Function call operator ( )
- Subscripting operator [ ]
- Class member access operator _ı

65. 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.

66. What is meant by inheritance?

Inheritance is the process by which objects of one class acquire the properties of another class. It supports the concept of hierarchical classification. It provides the idea of reusability. We can add additional features to an existing class without modifying it by deriving a new class from it.

67. What is meant by single inheritance?

A single class is derived from a single base class is called single inheritance.

Eg:

Base class

Derived class

Here class A is the base class from which the class D is derived. Class D is the public derivation of class B hence it inherits all the public members of B. But D cannot access private members of B.

68. What is multiple inheritance?

A class is derived from more than one base class, it is called multiple inheritance.

Eg: Base classes, Derived class

Here class C is derived from two base classes A & B.

69. What is hierarchical inheritance?

A number of classes are derived from a single base class then it is called hierarchical inheritance.

Eg : Hierarchical classification of students in University

70. What is multilevel inheritance?

If a class is derived from a class, which in turn is derived from another class, is called multilevel inheritance. This process can be extended to any number of levels.

Eg: Base class Grand father , Intermediate , Base class Father , Derived class Child

71. What is hybrid inheritance?

It is the combination of one or more types of inheritance.

• Multilevel
• inheritance
• Multiple
• inheritance

The class result will have both the multilevel and multiple inheritances.

72. What is meant by Abstract base class?

A class that serves only as a base class from which derived classes are derived. No objects of an abstract base class are created. A base class that contains pure virtual function is an abstract base class.

73. Write short notes on virtual base class.

A base class that is qualified as virtual in the inheritance definition. In case of multiple inheritance, if the base class is not virtual the derived class will inherit more than one copy of members of the base class. For a virtual base class only one copy of members will be inherited regardless of number of inheritance paths between base class and derived class.

Eg: Processing of students’ results. Assume that class sports derive the roll number from class student. Class test is derived from class Student. Class result is derived from class Test and sports. As a virtual base class As a virtual base class

74. Define Method overriding?

When a method in a subclass has the same name & type signature as a method in the subclass is said to override the method in the superclass. when an overridden method is called from within a subclass it will always refer to the version of that method defined by the subclass.

```
class A{
…….
Void show(){
Cout<<"Super class";
};
class B : A{
…….
Void show(){
Cout<<"sub class";
};
B sub;
Sub. show( );
        }
```

75. Define Scope Resolution operator .

Member functions can be defined within the class definition or separately using scope resolution operator (::). Defining a member function within the class definition declares the function inline, even if you do not use the inline specifier. Defining a member function using scope resolution operator uses following declaration

```
return-type class-name::func-name(parameter- list)
{
// body of function
}
```

Here the class-name is the name of the class to which the function belongs. The scope resolution, operator (::) tells the compiler that the function func-name belongs to the class class-name.

76. What is a Virtual Base Class?

The duplication of inherited members due to multipath avoided by making the common base class as virtual base class. When a class is made as virtual base class, then only one copy of that class will be inherited.

77. Define Pointer

A pointer is a variable that contains a memory address. Very often this address is the location of another object, such as a variable. For example, if x contains the address of y, then x is said to "point to" y. Pointer variables must be declared as such. The general form of a pointer variable declaration is

type *var-name;

Here, type is the pointer's base type. The base type determines what type of data the pointer will be pointing to. var-name is the name of the pointer variable.

78. What is This Pointer?

"**This pointer**" is a pointer that is automatically passed to any member function when it is called and it is a pointer to the object, that generates a call. (Or) "This" is a pointer and points to the object for which this function was called. Each time a member function is invoked it is automatically passed a pointer called "this" to the object that has invoked it.

79. Define Pure Virtual Function

A pure virtual function is a virtual function declared in a base class that has no definition relative to the base class. Such functions are called do nothing functions. Pure virtual function is a virtual function with no definition. They start with virtual keyword and ends with =0.

80. Define Abstract Base Class

Abstract class is a class which contains at least one pure virtual function in it. Abstract classes are used to provide an interface for its sub classes. Classes inheriting an Abstract class must provide definition to the pure virtual function; otherwise they will also become abstract class.

81. List out the Characteristics of Abstract Base Class.

- It cannot be instantiated, but pointers and references of Abstract class type can be created.
- It can have normal functions and variables along with a pure virtual function.
- They are mainly used for Up casting, so that its derived classes can use its interface.
- Classes inheriting an Abstract class must implement all pure virtual functions, or else they will become abstract too.

82. What is File?

File is the collection of related records stored in a disk.

83. What are the types of files?

There are three types :

- Sequential File
- Random File
- Indexed Sequential file

84. What is sequential file?

The records of the sequential files can be accessed one by one or sequentially.

85. What is indexed Sequential file?

The records of the index sequential file can be access randomly as well as sequentially

86. What is the name of Header file supports file classes?

The header file <fstream.h> contains all file supporting classes such as

```
ifstream in // input
ofstream out // output
fstream io // input and output
```

87. What is Random Access?

Random Access is the method of accessing a record from the file in any order.

88. What are the functions used for random accessing?

The random accessing can be done using the functions

Seekg()

Seekp()

89. What is the purpose of seekg() function?

The seekg() function is used to move get pointer(input) to a specified location.

Example:

Infile.seekg(10);

It moves the file pointer to the byte 10.

90. What is the purpose of seekp() function?

The seekp() is used to move the put pointer(output) to a specified location.

Example:

outfile.seekp(m) where m is the bytes.

91. What is Exception Handling?

- Exception Handing is a built-in mechanism to handle error during run time. It is more useful to manage and respond run time errors.
- The exception handling is built upon three keywords

a. Try

b. Catch

c. Throw

Syntax:-

```
Try {
// try block
}
catch(type1 arg) {
// catch blcok
}
```

92. Define Template.

- Templates are the foundation of generic programming, which involves writing code in a way that is independent of any particular type.
- A template is a blueprint or formula for creating a generic class or a function. The library containers like iterators and algorithms are examples of generic programming and have been developed using template concept.

93. Define STL.

- A set of general-purpose templatized classes for data structures and algorithms that could be used as a standard approach for storing and processing of data.
- The collection of these generic classes and functions is called the Standard Template Library.

94. Name the Components of STL.

The STL contains several components. These components work in conjunction with one another to provide support to a variety of programming solutions. They are:

A)Containers

B)Algorithms and

C)Iterators

95. Define Containers.

- A container is a way to store data, whether the data consists of built-in types such as int and float, or of class objects.
- The STL makes seven basic kinds of containers available, as well as three more that are derived from the basic kinds.
- Containers in the STL fall into two main categories: ***sequence*** **and** ***associative*****.**

96. Define Object-Oriented Analysis.

- Understanding the problem.
- Drawing the specification of requirement of the user and the software.
- Identifying the objects and their attributes.

- Identifying the services that each object is expected to provide (interface).
- Establishing inter-connections (collaborations) between the objects in terms of services.

97. Write the role of Software engineers.

- Software engineers have been trying various tools, methods, and procedures to control the process of software development in order to build high quality software with improved productivity.
- The methods provide "how to s" for building the software while the tools provide automated or semi-automated support for the methods.
- They are used in all the stages of software development process, namely, planning, analysis, design, development and maintenance.
- The software development procedures integrate the methods and tools together and enable rational and timely development of software systems.

98. Components of Software development.

- Software engineers have been trying various tools, methods, and procedures to control the process of software development in order to build high quality software with improved productivity.
- The methods provide "how to s" for building the software while the tools provide automated or semi-automated support for the methods.
- They are used in all the stages of software development process, namely, planning, analysis, design, development and maintenance

99. Procedure-Oriented Paradigm.

- Software development is usually characterized by a series of stages depicting the various asks involved in the development process.
- The classic life cycle is based on an underlying model, commonly referred to as the "water fall" model.
- This model attempts to break up the identifiable activities into series of actions, each of which must be completed before the next begins.
- The activities include problem definition, requirement analysis, design, coding, testing, and maintenance.

- Further refinements to this model include iteration back to the previous stages in order to incorporate any changes or missing links

100. Define Containers.

- A container is a way to store data, whether the data consists of built-in types such as int and float, or of class objects.
- The STL makes seven basic kinds of containers available, as well as three more that are derived from the basic kinds.
- Containers in the STL fall into two main categories: ***sequence* and *associative*.**
- The sequence containers are ***vector, list,* and *deque.***

101. What is Sequence Containers?

- A sequence container stores a set of elements in what you can visualize as a line, like houses on a street. Each element is related to the other elements by its position along the line. Each element is preceded by one specific element and followed by another.
- The STL provides the *vector* container to avoid these difficulties. This can be very time consuming.
- The STL provides the *list* container, which is based on the idea of a linked list.
- The third sequence container is the *deque*, which can be thought of as a combination of a stack and a queue. Both input and output take place on the top of the stack.
- A queue, on the other hand, uses a first-in-first-out arrangement: data goes in at the front and comes out at the back, like a line of customers in a bank.
- A deque combines these approaches so you can insert or delete data from either end. The word deque is derived from Double-Ended QUEue. It’s a versatile mechanism that’s not only useful in its own right, but can be used as the basis for stacks and queues.

102. What is Associative Containers?

- An associative container is not sequential; instead it uses *keys* to access data. The keys, typically numbers or stings, are used automatically by the container to arrange the stored elements in a specific order.
- It’s like an ordinary English dictionary, in which you access data by looking up words

arranged in alphabetical order and the container converts this key to the element's location in memory.

- There are two kinds of associative containers in the STL: ***sets* and *maps*.**
- These both store data in a structure called a ***tree***, which offers fast searching, insertion, and deletion.

103. Define String.

It is collection of characters.

104. List out the functions supported by String Class

- append(): This function appends a part of a string to another string
- assign():This function assigns a partial string
- at(): This function obtains the character stored at a specified location
- begin(): This function returns a reference to the start of the string
- capacity(): This function gives the total element that can be stored
- compare(): This function compares a string against the invoking string
- empty(): This function returns true if the string is empty
- end(): This function returns a reference to the end of the string
- erase(): This function removes character as specified
- find(): This function searches for the occurrence of a specified substring
- length(): It gives the size of a string or the number of elements of a string
- swap(): This function swaps the given string with the invoking one
- *portant Constructors obtained by String Class*
- String(): This constructor is used for creating an empty string

105. Give the functions that manipulate null-terminated strings.

- strcpy(str1, str2): Copies string str2 into string str1.
- strcat(str1, str2): Concatenates string str2 onto the end of string str1.
- strlen(str1): Returns the length of string str1.
- strcmp(str1, str2): Returns 0 if str1 and str2 are the same; less than 0 if str1<str2; greater than 0 if str1>str2.
- strchr(str1, ch): Returns a pointer to the first occurrence of character ch in string str1.
- strstr(str1, str2): Returns a pointer to the first occurrence of string str2 in string str1.