Model Viva Questions for “**Object Oriented Programmings in C++ Lab**”

Common to: **CS 3rd sem & IT 3rd sem**

1. Program showing the use of data type and basics of C++.

(a) Program to calculate sum of two numbers.

(b) Program to calculate simple interest.

Q1: What are literals?
A1: Literals are data items that never change their value during a program run.

Q2: What are five fundamental data types in C++?
A2 : char, int, float, double and void

Q3: What is reference?
A3**:** Reference is an alternative name for a variable.

Q4: What is a variable?
A4**:** A variable is a named storage location. It has two associated values : rvalue that gives its contents and lvalue that gives its address.

Q5: What does unary + and unary – gives?
A5**:** Unary + gives the value of its operand , unary – changes the sign of its operand’s value.

Q6: What is the function of postfix version of increment and decrement?
A6: The postfix version first uses the value of its operand and then changes(increments or decrements) it i.e. it follows use-then-change rule.

Q7: What is the function of prefix version of increment and decrement?
A7: The prefix first changes (increments or decrements) its operand’s value and then uses it i.e. , it follows change-then-use rule.

Q8: What is the thruth value of non –zero expression?
A8: The truth value of non –zero expression is 1 i.e. , true and for zero expression , it is 0 i.e., false.

Q9: What type of expression is arithmetic expression?
A9**:** Arithmetc expression can either be integer expression or real expressions or mixed mode expressions.

Q10: What we do in implicit conversion ?
A10: In implicit conversion, all operands are converted upto the type of the larget operand , which is called type promotion.

Program showing C++ programs based on object and classes :

(a) Program showing arrays within a class.

(b) Program containing arrays of object.

Q1: What is class?
A1: A class is an expanded concept of a data structure: instead of holding only data, it can hold both data and functions.

Q2: What are the ways to define a class?
A2: While defining a class , its member functios can be either defined within the class definition or outside the class definition.

Q3: Are the inline functions called?
A3: Inline functions are not called , their code replaces their function –calls in the program.

Q4: What does a class support?
A4: A class supports Oops features encapsulation by hiding data and associated function together.

Q5: What is nesting of member function ?
A5: When a member function is called by another member function of the same class, it is called nesting of member functions.

Q6: How are objects created?
A6: Objects can be created separately to store their data members.

Q7: What are static member functions?
A7: The static member functions re the functions that can access only the static members.

Q8: What are static data members?
A8: The static data members are global variabes for its class type objects . they are commonly shared by all the objects of that class type.

Q9: What is friend function?
A9: A fiend function is a function which is not a member of a class but which is given special permission to access private and protected members of the class.

Q10: How are class members referenced?
A10: The class members are referenced using objects of the class and the dot operator. For instance , to access member x of an object A of class class ABC we'll give A.x

At least one C++ program based on the following :

(a) Constructors and destructors-Program to generate a series of Fibonacci numbers using constructor and destructor.

(b) Overloading unary operator-Program to generate a Fibonacci series by overloading a postfix operator.

(c)Overloading binary operator- Program to perform overloading of a plus operator for finding the sum of two given class objects.

Q1: What is constructor?
A2 :A constructor is a member function with the same nme as that of its class and it is used to initialize the objects of that class type with a legel initial value.

Q2: If a class has no constructor, what does a compiler do?
A2 : If a class has no constructor defined , the compiler automatically generates one.

Q3: What is default constructor?
A3: Constructor with no arguments or all the arguments has default values.

Q4: What is the difference between a copy constructor and an overloaded assignment operator?
A4: A copy constructor constructs a new object by using the content of the argument object. An overloaded assignment operator assigns the contents of an existing object to another existing
object of the same class.

Q5: What is destructor?
A5: A destructor is a member function with same name as that of its class but preseded by a tilde (~).

Q6: How is the argument of copy constructor passed?
A6: The argument to a copy constructor is always passed by reference.

Q7: How are the constructors called?
A7: Constructors may be called implicitly as well as explicitly.

Q8: Can a constructor throws an exception? How to handle the error when the constructor fails?
A8: The constructor never throws an error.

Q9: Can a copy constructor accept an object of the same class as parameter, instead of reference
of the object?
A9: No. It is specified in the definition of the copy constructor itself. It should generate an error if a programmer specifies a copy constructor with a first argument that is an object and not a
reference.

Q10: How can I handle a constructor that fails?
A10: throw an exception. Constructors don't have a return type, so it's not possible to use return codes. The best way to signal constructor failure is therefore to throw an exception.

C++ program based on the following :

(a) Inheritance- Program to demonstrate how ambiguity is avoided in single inheritance using scope resolution operator.

(b) Multiple Inheritance- Program to demonstrate how ambiguity is avoided in multiple inheritance using scope resolution operator.

Q1: What is inheritance?
A1: Inheritance allows one class to reuse the state and behavior of another class. The derived class inherits the properties and method implementations of the base class and extends it by overriding methods and adding additional properties and methods.

Q2: What is multiple inheritance(virtual inheritance)? What are its advantages and disadvantages?
A2: Multiple Inheritance is the process whereby a child can be derived from more than one parent class. The advantage of multiple inheritance is that it allows a class to inherit the functionality of more than one base class thus allowing for modeling of complex relationships. The disadvantage of multiple inheritance is that it can lead to a lot of confusion(ambiguity) when two base classes implement a method with the same name.

Q3: What does inheritance supports?
A3: Inheritance Supports reusability of code and is able to simulate the transitive nature of real life objects.

Q4: What are the different forms of inheritance?
A4: Inheritance has many forms: single inheritance, multiple inheritance , hierarchical inheritance, multilevel inheritance, hybrid inheritance.

Q5: How can we inherit private member of a base class?
A5: To make a private member of a base class inheritable , declare it under protected section of base class.

Q6: What occurs in publicly derived class ?
A6: In a publicly derived class , the public and protected members remain public and protected.

Q7: What occurs in privately derived class?
A7: In a privately derived class , the public and protected members of the base class becomes private members .

Q8: What is single inheritance?
A8: When a class inherits from a single base class , it is single inheritance .

Q9: For what is the derived class constructor responsible?
A9: The derive class constructor is responsibe for invoking (and passing arguments to) the base class constructor.

Q10: When is a template a better solution than a base class?
A10: When you are designing a generic class to contain or otherwise manage objects of other types, when the format and behavior of those other types are unimportant to their containment or management, and particularly when those other types are unknown (thus, the generality) to the
designer of the container or manager class

One C++ program based on the following:

(a)Polymorphism-Program to illustrate the dynamic binding of member function of a class.

(b) Overloading- Program to illustrate function overloading.

(c) Overriding- Program to show method overriding.

Q1: What is Polymorphism?
A1: Polymorphism allows a client to treat different objects in the same way even if they were created from diffferent classes and exhibit different behaviors. You can use implementation inheritance to achieve polymorphism in languages such as C++ and Java. Base class object's pointer can invoke methods in derived class objects. You can also achieve polymorphism in C++ by function overloading and operator overloading.

Q2: What are the types of Polymorphism?
A2: There are two types of Polymorphism namely , compile time polymorphism and runtime polymorphism.

Q3: What is problem with Runtime type identification?
A3: The run time type identification comes at a cost of performance penalty. Compiler maintains the class

Q4: What is Operator overloading?
A4: When an operator is overloaded, it takes on an additional meaning relative to a certain class. But it can still retain all of its old meanings.
Examples:
1) The operators >> and << may be used for I/O operations because in the header, they are overloaded.
2) In a stack class it is possible to overload the + operator so that it appends the contents of one stack to the contents of another. But the + operator still retains its original meaning relative to other types of data.

Q5: What does **this** pointers points to?
A5: **this** pointer points to the object for which this function was called.

Q6: What is Overriding?
A6: To override a method, a subclass of the class that originally declared the method must declare a method with the same name, return type (or a subclass of that return type), and same parameter list.
The definition of the method overriding is:
· Must have same method name.
· Must have same data type.
· Must have same argument list.
Overriding a method means that replacing a method functionality in child class. To imply overriding functionality we need parent and child classes. In the child class you define the same method signature as one defined in the parent class.

Q7: Why do we need virtual function?
A7: When a function is made virtual, C++ determine which function to use at runtime based on the of object pointed to by the base pointer, rather than the type of pointer.

Q8: How is runtime polymorphism achieved?
A8: Run time polymorphism is achived only when a virtua l function is accessed through a pointer to base class. It cannot be achieved using object name along with the dot operator to access virtual function.

Q9: Can we have a virtual constructor?
A9: We can have a virtual destructor but not virtual constructor.

Q10: How is object pointer useful?
A10: Object pointers are useful in creatng objects at run time. It can be used to access the public members of an object, along with an arrow operator.

Some C++ program should be conducted on each of the following:

- **(a) 2d array sorting- Program to sort 2D array using linear representation of the array.**
- **(b) String manipulation-Program shows how to create a string that contains the full name from first name and last name (e.g., firsname = "John", lastname = "Smith", fullname = "Smith, John"):**
- **(c) Pointer to objects – Program to illustrate the use of pointer to objects.**
- **(d) Use of this Pointer- Program to illustrate the use of this pointer.**
- **(e) Pointers to derived class- Program to illustrate the use of pointers to derived class**

Q1: What is a dangling pointer?
A1: A dangling pointer arises when you use the address of an object after its lifetime is over. This may occur in situations like returning addresses of the automatic variables from a function or using the address of the memory block after it is freed.

Q2: What is Memory Leak?
A2: Memory which has no pointer pointing to it and there is no way to delete or reuse this memory(object), it causes Memory leak.

```
{
Base *b = new base();
}
```

Out of this scope b no longer exists, but the memory it was pointing to was not deleted. Pointer b itself was destroyed when it went out of scope.

Q3: What issue do auto_ptr objects address?
A3: If you use auto_ptr objects you would not have to be concerned with heap objects not being deleted even if the exception is thrown.

Q4: Is there any problem with the following : char*a=NULL; char& p = *a;?
A4: The result is undefined. You should never do this. A reference must always refer to some object.

Q5: What is the difference between a pointer and a reference?
A5: A reference must always refer to some object and, therefore, must always be initialized; pointers do not have such restrictions. A pointer can be reassigned to point to different objects while a reference always refers to an object with which it was initialized.

Q6: What is the difference between const char *myPointer and char *const myPointer?
A6: Const char *myPointer is a non constant pointer to constant data; while char *const myPointer is a constant pointer to non constant data.

Q7: From which is the pointer to object of a base class type compatible ?
A7: Pointers to object of a base class type is compatible with pointer to onject of a derived class . Therefore , we can use single pointer variable to point to objects of base class as well as derived class.

Q8: What is a **this** pointer?
A8: A special pointer known as **this** pointer stores the address of the object that is currently invoking a member function.

Q9:How are objects passed to functions?
A9: Objects can be passed to functions through call-by –value as well as call-by- reference mechanism.

Q10: Why is Arrow operator (“-> “) used?
A10: The arrow operator is used to access the public members of the class with a pointer to an object.

Title of the Practical: At least two program based on file handling.
(a)Program using getline() and write() functions.
(b) Program showing read and write operations.

Q1: What is a file?
A1: A file is a bunch of bytes stored on some storage device.

Q2: What does the fstream class do?
A2: The **fstream** class ties a file to the input stream for input; **ofstream** class ties a file to output stream for output ; and **fstream** class ties a file to the stream for both input and output

Q3: How a file is closed?
A3: A file is closed i.e. its connection with the stream is terminated using function close().

Q4 What are Templates ?
A4:C++ Templates allow u to generate families of functions or classes that can operate on a variety of different data types, freeing you from the need to create a separate function or class for each type. Using templates, u have the convenience of writing a single generic function or class definition, which the compiler automatically translates into a specific version of the function or class, for each of the different data types that your program actually uses. Many data structures and algorithms can be defined independently of the type of data they work with. You can increase the amount of shared code by separating data-dependent portions from data-independent portions, and templates were introduced to help you do that.

Q5. Write two member functions belonging to fstream class.
A5: get(), seekg, seekp().

Q6: Is What is the difference between read() and write() ?
A6:The read () lets one to read a unit of information fron a file and a write () lets one to write a unit of information to a file

Q7: Name the stream class supported by C++ for file input and output.
A7: ofstream , ifstream, fstream

Q8: What is stream? Name the stram generally used for I/O.
A8: A stream is a sequence of bytes. Or in other words , a stream is a flow of bytes into or out of a program. Generally three streams are used for fle I/O. These are:
(a) ifstream
(b) ofstream
(c)fstream

Q9. What are manipulators?
A9:Manipulators are special functions that can be included in the I/o statement to alter the format parameters of stream.

Q10: Name different manipulators.
A10:setw(), setprecisin(), setfill(), setiosflags(), resetiosflags().

Title of the Practical: At least four C++ prgrams based on Graphics function.

(a)Progarm to draw a line.

(b) Program to draw a circle.

(c) Program to draw a rectangle.

(d) Program to draw a polygon.

Q1: What is BGI?
A1:The Borland graphics Interface(BGI) is collection of function and procedure that are often called as graphics toolkit which an application program may use to solve specifc graphics problems.

Q2: Name diff erent Graph initialization and closing routines.
A2: closegraph() getmodename(), detectgraph(), initgraph(), setgraphmode(), etc.

Q3: What is the function of initgraph() ?
A3: The initgraph() method is used to initialize the graphics system and to put yhe hardware into graphics mode.

Q4: What is the function of closegraph() ?
A4: The closegraph() is a method that is used to shut down the graphics system.

Q5: Write the syntax for drawing a circle through circle() .
A5: The syntax of circle() is:

void far circle(int x , int y, int radius);

Q6: What is the function of setfillpattern()?
A6: The setfillpattern() is a method that is used to select a user- defined fill pattern.
The general syntax of setfillpattern() is:

void far setfillpattern(char far +upattern , int color);

Q7: What header file is used to use graphics in C++..
A7: Graphics.h.

Q8: Why is lineto() method used?
A8: The lineto() is a method that is used to draw a lne from the current pointer to(x,y) , The general syntax of lineto () is:

void far lineto(int x, int y);

Q9: What is the general syntax for initgraph() ?
A9: The general syntax for initgrap() is:

void far initgraph(int far *graphdriver, int far *graphmode, char far *pathtodriver);

Q10: What is the general syntax for setlinestyle?
A10: The general syntax for setlinestyle is:

void far setlinestyle(int linestyle, unsigned upattern, int thickness);