Basics of C++ and Object Oriented Programming

Self-Assessment Questions

Find the answers to the following questions:

1. What is object oriented programming? How is it different from procedure-oriented programming?

Ans. Object-oriented programming (OOP) is a style of programming characterized by the identification of classes of objects closely linked with the methods (functions) with which they are associated. It also includes ideas of inheritance of attributes and methods.

As compared to object-oriented programming, procedural programming is less secure. Procedural programming follows a top-down approach during the designing of a program. It gives importance to the concept of the function and divides the large programs into smaller parts or called as functions. Procedural programming is straightforward. Unlike object-oriented programming, there are no access modifiers introduced in procedural programming.

1. What are the benefits of Object Oriented Programming?

OOP language allows to break the program into the bit-sized problems that can be solved easily (one object at a time). The new technology promises greater programmer productivity, better quality of software and lesser maintenance cost. OOP systems can be easily upgraded from small to large systems.

Advantages of OOPS:

1. Modularity for easier troubleshooting
2. Reuse of code through inheritance
3. Flexibility through polymorphism
4. Effective problem solving
5. Explain the following concepts of Object Oriented Programming briefly?

* Objects:

An object in OOPS is nothing but a self-contained component which consists of methods and properties to make a particular type of data useful. For example color name, table, bag, barking. When you send a message to an object, you are asking the object to invoke or execute one of its methods as defined in the class.

* Classes:

In object-oriented programming, a class is a blueprint for creating objects (a particular data structure), providing initial values for state (member variables or attributes), and implementations of behavior (member functions or methods). The user-defined objects are created using the class keyword.

* Data Abstraction and Encapsulation:

Abstraction is a feature of OOPs that hides the unnecessary detail but shows the essential information. Encapsulation is also a feature of OOPs. It hides the code and data into a single entity or unit so that the data can be protected from the outside world. It solves an issue at the design level.

* Inheritance:

Inheritance in OOP = When a class derives from another class. The child class will inherit all the public and protected properties and methods from the parent class. In addition, it can have its own properties and methods. An inherited class is defined by using the extends keyword.

* Polymorphism:

Polymorphism is one of the core concepts of object-oriented programming (OOP) and describes situations in which something occurs in several different forms. In computer science, it describes the concept that you can access objects of different types through the same interface.

* Dynamic Binding:

Dynamic binding refers to linking a procedure call to code that will execute only once. The code associated with the procedure is not known until the program is executed, which is also known as late binding.

1. What are promising areas of applications of OOP?

Main application areas of OOP are:

User interface design such as windows, menu.

Real Time Systems. Simulation and Modeling.

Object oriented databases.

AI and Expert System.

Neural Networks and parallel programming.

Decision support and office automation systems etc.

1. What is object-based approach? How does object-oriented approach differ from object-based approach?

In the object-oriented approach, the focus is on capturing the structure and behavior of information systems into small modules that combines both data and process.

Object Oriented languages support all the features of OOPS but object based languages do not support all the features of OOPS. Object oriented languages support inheritance but object based languages do not support inheritance.

1. How does a main function in C++ differ from main() in C?

Main function of C may be void, when returns nothing. In C++ main can not be void. It will return int value. In c main function, we do declare all variables together in beginning of the program.

1. What is structure of C++ program?

In C++, a program is divided into the following three sections: Standard Libraries Section. Main Function Section. Function Body Section.

1. What are new data types introduced in C++? (bool, wchar\_t)

A boolean data type is declared with the bool keyword and can only take the values true or false. When the value is returned, true = 1 and false = 0.

wchar\_t is similar to char data type, except that wide char take up twice the space and can take on much larger values as a result. char can take 256 values which corresponds to entries in the ASCII table. On the other hand, wide char can take on 65536 values which corresponds to UNICODE values which is a recent international standard which allows for the encoding of characters for virtually all languages and commonly used symbols.

1. Explain briefly new keywords in C++ (friend,mutable,explicit,using, typename)

Friend Class A friend class can access private and protected members of other class in which it is declared as friend. It is sometimes useful to allow a particular class to access private members of other class.

The mutable storage class specifier in C++ (or use of mutable keyword in C++) auto, register, static and extern are the storage class specifiers in C. typedef is also considered as a storage class specifier in C. C++ also supports all these storage class specifiers. In addition to this C++, adds one important storage class specifier whose name is mutable.

Explicit Keyword in C++ is used to mark constructors to not implicitly convert types in C++. It is optional for constructors that take exactly one argument and work on constructors(with a single argument) since those are the only constructors that can be used in typecasting.

​The using keyword is used to: Bring a specific member from the namespace into the current scope. Bring all members from the namespace into​ the current scope. Bring a base class method ​or variable into the current class’s scope.

" typename " is a keyword in the C++ programming language used when writing templates. It is used for specifying that a dependent name in a template definition or declaration is a type.

1. Discuss some new operators in C++ (::, new, delete, endl)

The scope resolution operator ( :: ) is used for several reasons. For example: If the global variable name is same as local variable name, the scope resolution operator will be used to call the global variable. It is also used to define a function outside the class and used to access the static variables of class

.

Use of the new operator signifies a request for the memory allocation on the heap. If the sufficient memory is available, it initializes the memory and returns its address to the pointer variable. The new operator should only be used if the data object should remain in memory until delete is called.

Delete is an operator that is used to destroy array and non-array(pointer) objects which are created by new expression. New operator is used for dynamic memory allocation which puts variables on heap memory

Endl in C++ is a manipulator or in simple terms a command. So when endl is encountered, the operating system will flush the output buffer and insert a new line. As you would recall, flushing of the buffers is done periodically without the need of an explicit flush call to the buffer.

1. What are the benefits of new/delete vs malloc/dealloc?

The main difference between new and malloc is that new invokes the object's constructor and the corresponding call to delete invokes the object's destructor. There are other differences: new is type-safe, malloc returns objects of type void\* new throws an exception on error, malloc returns NULL and sets errno new is an operator and can be overloaded, malloc is a function and cannot be overloaded new[], which allocates arrays, is more intuitive and type-safe than malloc malloc-derived allocations can be resized via realloc, new-derived allocations cannot be resized malloc can allocate an N-byte chunk of memory, new must be asked to allocate an array of, say, char types

1. What is difference between structure in C and structure in C++?

Structures in C, cannot have member functions inside structures. Structures in C++ can hold member functions with member variables. We cannot initialize the structure data directly in C. We can directly initialize structure data in C++.

1. What is the difference between structure and class in C++?

The only difference between a struct and class in C++ is the default accessibility of member variables and methods. In a struct they are public; in a class they are private.

1. What is a reference variable? What is most significant advantage of using references instead of pointers?

In C++, a reference variable is a variable that is used to refer to an existing variable. This type of variable is created using the & operator.

Reference variables are cleaner and modish as compare to the pointers; they can also be used while passing in the function as arguments, known as call by references

1. What are inline functions? When would you make a function inline?

Inline function is a function that is expanded in line when it is called. When the inline function is called whole code of the inline function gets inserted or substituted at the point of inline function call. This substitution is performed by the C++ compiler at compile time.

We will make a function inline when the functions are small that called often. Inline functions run a little faster than the normal functions as the compiler replaces the function call statement with the function code itself and then compiles the entire code.

1. When do we use default arguments in a function?

The default arguments are used when you provide no arguments or only few arguments while calling a function. The default arguments are used during compilation of program.

1. What do we mean by overloading of a function? How does a compiler distinguishes between a set of overloaded functions having the same name?

Function overloading is a feature of object oriented programming where two or more functions can have the same name but different parameters.

The compiler distinguishes overloaded methods by their signatures—a combination of the method's name and the number, types and order of its parameters, but not its return type. If the compiler looked only at method names during compilation,

1. What are static members (data and functions) in class? Justify the need of static members.

The static member functions are special functions used to access the static data members or other static member functions. A member function is defined using the static keyword. A static member function shares the single copy of the member function to any number of the class' objects.

A typical use of static members is for recording data common to all objects of a class. For example, you can use a static data member as a counter to store the number of objects of a particular class type that are created.

1. What is mutable data member? What is its significance?

Mutable data members are those members whose values can be changed in runtime even if the object is of constant type. It is just opposite to constant. Sometimes logic required to use only one or two data member as a variable and another one as a constant to handle the data.

The mutable keyword allows the data member of a class to change within a const member function. - It allows to assign the values to a data member belonging to a class defined as “Const” or constant. - It allows a const pointer to change members.

1. What is a friend function? What are its merits and demerits?

A friend function in C++ is defined as a function that can access private, protected and public members of a class. The friend function is declared using the friend keyword inside the body of the class.

merits: we can able to access the other class members in our class if,we use friend keyword. we CAN access the members without inheriting the class. demerits: Maximum size of the memory will occupied by objects according to the size of friend Members. we cant do any run time ploymorphism concepts in those members.

1. What is a nested class? What is its use?

If a class is defined inside another class, the inner class is termed as nested class. The inner class is local to the enclosing class.

A nested class is a class that is declared in another class. The nested class is also a member variable of the enclosing class and has the same access rights as the other members. However, the member functions of the enclosing class have no special access to the members of a nested class.