1. What is a class?

The class is the single most important C++ enhancement for implementing abstraction, encapsulation, and data hiding and tying them together.

A *class* is a C++ vehicle for translating an abstraction to a user-defined type. It combines

data representation and methods for manipulating that data into one neat package.

1. How does a class accomplish abstraction, encapsulation, and data hiding?

An abstraction is a simplified view of an object in the user’s own vocabulary In C++ class is used through its access control mechanism to implement the abstract interface In OO and C++, an abstraction is the simplest interface to an object that provides all the features and services the intended users expect.

Car designer divide it to number of logical entities (engine, transmission, clutch..).

• Each part does not know the details of other parts ***(encapsulation)***. He knows only how he can use them.

1. What is the relationship between an object and a class?

An object is a software entity that combines state and behavior.

A class is a programming construct that defines the common state and behavior of a group of similar objects

Classes & objects A class has a name, and it describes the state (member data) and services (member functions) provided by objects that are instances of that class. The runtime system creates each object based on a class definition.

1. In what way, aside from being functions, are class function members different from class data members?

A member function of a class is a function that has its definition or its prototype within the class definition like any other variable. It operates on any object of the class of which it is a member, and has access to all the members of a class for that object. While data members are the variables that are declared in a member specification of a class.

1. When are class constructors called? When are class destructors called?

Constructors are like "init functions". Minimally they initialize internally used fields. They may also allocate resources (memory, files, semaphores, sockets, etc).  Every time an instance of a [class](http://www.cprogramming.com/tutorial/lesson12.html) is created the constructor method is called. The constructor has the same name as the class and it doesn't return any type, while the destructor's name it's defined in the same way, but with a '~' in front. If you create an automatic storage class object, as the examples have been doing, its destructor is called automatically when the program exits the block of code in which the object is defined

1. What is a default constructor? What is the advantage of having one?

A default constructor is a constructor that is used to create an object when you don’t provide explicit initialization values and is called with no arguments.

1. What are this and \*this?

this is a pointer used to point an already defined object when I need to call the function from main while \*this is a considered as the object itself which is already defined at main and used to call the function .