1. What is a class?

What is an Object first? An Object is a thing that has properties and functions (i.e. a pen, a purse, a desk, a car or even a person) it might be a virtual thing too like an alien or a student account on facebook.

Now we can classify each group of objects that have common characteristics (or properties) and have the same generic or specific functions into Classes.

So classes alone aren’t any real thing in our system, though they are the source, the classification from which the real things (Objects) are generated (or instantiated), then customized (red pen, blue pen. Volvo, BMW... etc.)

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

Abstraction is giving the user the ability to deal with the Object/Class from Outside, without any further needs to get into details.

Encapsulation is the protection of the Object's properties and functions by putting a force field around them like a medicine Capsule. You may see what's inside of a pill (if its cover is transparent) but you can't change its contents.

We need encapsulation because simply our variables are based on assumptions that the user may violate if it was wide open for him to change.

Data Hiding is achieved by setting variables as private. (Hides them from the class user, will appear in debugging though)

A Soda Can class with abstraction only, without encapsulation would have attributes like the drink name and the can color as public and the ingredients of the drink itself as private.

If Encapsulation was added the can color and name would be protected from being changed by the user (buyer) by setting them as private and declaring setters and getters for them to make sure that the user doesn’t set an invalid name to the can for example.

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

An Object is an instance of the class, a Class is a Classification of an Object (if Objects were Candies, Classes would be the Candy Recipe) and the user who uses the recipe to create a new Candy is the factory using the virtual machine, Compiler and so whatever used as machines.

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

Function members are what an Object instantiated from a Class Can do. On the other hand, data members are what that Object has or what the Object is.

For a Car class: color, seats, doors… etc. are data members

While steers right & left, goes forward and backward & stops are function members.

5- Define a class to represent a bank account. Data members should include the depositor’s name, the account number (use a string), and the balance.

Member functions should allow the following:

\* Creating an object and initializing it.

\* Displaying the depositor’s name, account number, and balance

\* Depositing an amount of money given by an argument

\* Withdrawing an amount of money given by an argument

Just show the class declaration, not the method implementations.

Class BankAccount{

Private:

String depositerName, accountNo;

int balance;

void updateBalance(int amount);

Public:

//Constructor for initialization

BankAccount( String depName, String AccountNo, int Balance);

//Getters for all

String getDepositerName();

String getAccountNo();

int getBalance();

//Depositing and Withdrawing an amount of money

//Both does some checks then call the private //"updateBalance" Method

void deposit(int depositAmount);

void withdraw(int withdrawalAmount);

};

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

Constructors are called once for the one Object when it is instantiated.

Destructors are called when the pointers to the Objects are deleted (depends whether the pointers are smart ones or not though)

Or when the Object is defined locally in some block and the compiler exits the block, the Object then becomes out of Scope and should be deleted from the stack, so its destructor is called.

1. Provide code for a constructor for the bank account class from Chapter Review Question 5.

//Constructor for initialization

BankAccount( String depName, String AccountNo, int Balance){

this.depositeName = depName;

this.accountNo = AccountNo;

this.balance = Balance;

}

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

Default Constructor is a Constructor with no arguments. It may be compiler-generated or written by the developer. Either ways its advantage differs due to the class type.

* In general it helps execute some code @ instantiation of an Object, like initializing its attributes, or counting the instantiated Objects into a static variable, or so whatever.
* If the class has a base class (super class) the default constructor calls the constructor of the base class too.
* If the fields of the class has constructors, they'll be called too. (non-POD fields)
* Initializes the vtable pointers to help achieve polymorphism in polymorphic classes (like when we want to call a method of a sub-class which was passed by the type of its parent class).

1. J

10- What are this and \*this?

this is a pointer to the address of the current object, \*this is either a reference to or a clone of the Object.