Chapter 10 Review Questions

Asmaa Nabil
BN: 33716

(1) What is a class?

A class is an object oriented C++ vehicle for translating an abstraction to a user-defined type. It combines data members (variables) and member functions (methods) for manipulating that data into one neat package.

(2) How does a class accomplish abstraction, encapsulation, and data hiding?

- Data abstraction is a programming (and design) technique that relies on the separation of interface and implementation. C++ classes provides great level of data abstraction. They provide sufficient public methods to the outside world to play with the functionality of the object and to manipulate object data, i.e., state without actually knowing how class has been implemented internally.
- Encapsulation is an Object Oriented Programming concept that binds together the data and functions that manipulate the data, and that keeps both safe from outside interference and misuse. C++ supports the properties of encapsulation through the creation of user-defined types, called classes.
- Data hiding is one of the important aspects of a class. Data hiding means to hide the members of the class and providing the access to only some of them. We can make the members of the class private or public. In private, the outside world cannot access those members which been made private. And rest we can make public. Only the public members are accessed by the outside world and all the private members can be accessed using only the public members. In this way a class provides security to its data members.

(3) What is the relationship between an object and a class?

A class defines the properties and behavior for the objects represented by the abstraction. Abstraction is a property of object oriented programming. It denotes the essential properties and behaviors of an object. It hides code and data. A class thus denotes a category of objects and act as a blueprint for creating such objects. An object exhibits the property and behaviors defined by its class. Generally, an object is an instance of a class.

(4) In what way, aside from being functions, are class function members different from class data members?

The work that a class does is performed by its member functions. The state that it maintains is stored in its data members.

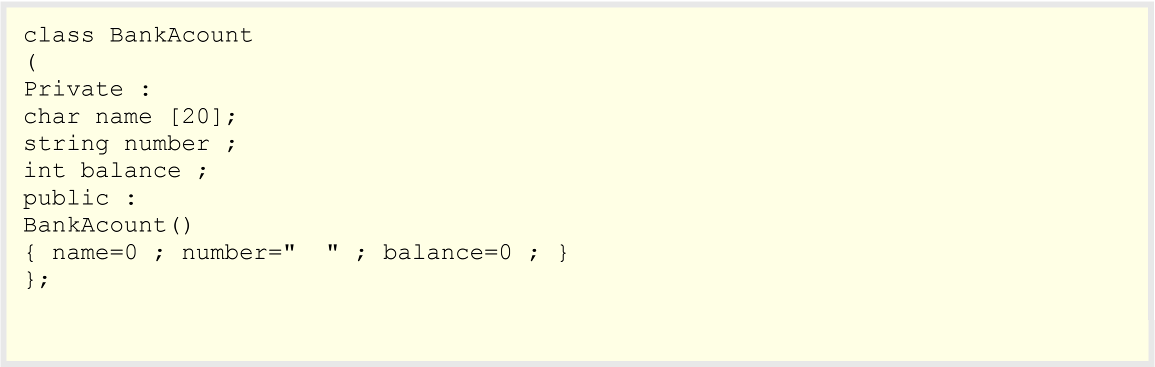
(5) Define a class to represent a bank account.

```
Class BankAcount
{
Private  :
char name[20];
string number;
int balance;
public :
void OpenBalance();
void dislpay();
void deposit(int dep);
void withdraw(int wd);
};
```

(6) When are class constructors called? When are class destructors called?

These constructors are called automatically at the appearance of a new object for the first time only. The destructor is called after the finalisation section of the unit it is declared in.

(7) Provide code for a constructor for the bank account class.



```
class BankAcount
(
Private :
char name [20];
string number ;
int balance ;
public :
BankAcount()
{ name=0 ; number="  " ; balance=0 ; }
};
```

(8) What is a default constructor? What is the advantage of having one?

It is a constructor that can be called without having to provide any arguments, irrespective of whether the constructor is auto-generated or user-defined. The advantage is that a constructor with formal parameters can still be called without arguments if default arguments were provided in the constructor's definition.

(9) What are this and *this?

Each member function, including constructors and destructors, has a 'this' pointer. The special property of the 'this' pointer is that it points to the invoking object. If a method needs to refer to the invoking object as a whole, it can use the expression *this.