Question 1

(a) Complete the code based on the instructions given in the comments:

```
#include<iostream>
using namespace std;

class MasterStudent
{   string name, title;
    int status;

    public:
     MasterStudent(string n, string t, int x)
     {   name = n;
         title = t;
         status = x;
     }

     MasterStudent()
     { name = "Peter";
       title = "A Study on the Usability Factors of Mobile Apps.";
       status = 1;
     }

     int getStatus()
     { return status;
     }
     string getName()
     { return name;
     }
     string getTitle()
     { return title;
     }

     ~MasterStudent()
     { cout<<"\n\n~End of Details~Student~"<<name<<endl;
     }
};

int main()
{  MasterStudent MS1;
   cout<<"================================="<<endl;
   cout<<"   Masters Student Details       "<<endl;
   cout<<"================================="<<endl;
   cout<<"Name \t: "<<MS1.getName()<<endl;
   cout<<"Title \t: "<<MS1.getTitle()<<endl;
   cout<<"Status \t: ";
   if(MS1.getStatus() ==1) cout<<"Approved"<<endl;
   else cout<<"Pending"<<endl;

   //Create another object passing the values
   //"Aliana Mahmud" for name,
   //"Customer Satisfaction towards Green Products" for title, 0 for status
   //Use the cout statements given earlier to display the content of the
   //new object.
   //Refer to the part labeled X in the sample output for the output to be
   //displayed.
```

```
}
```

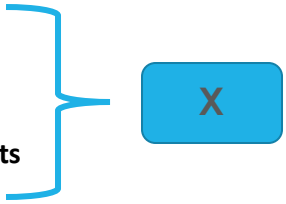
Sample Output Screen 1 a)

```
==============================
  Masters Student Details
==============================
Name    : Peter
Title   : A Study on the Usability Factors of Mobile Apps.
Status  : Approved

==============================
  Masters Student Details
==============================
Name    : Aliana Mahmud
Title   : Customer Satisfaction towards Green Products
Status  : Pending


~End of Details~Student~Aliana Mahmud


~End of Details~Student~Peter
```



b) Modify the main() of program at (a) to initialize an ***array of 4 objects of MasterStudent*** class with the data in the table below and display the output (refer to sample output screen 1b) :

Name	Title	Status
Philip Morales	"Working with Generation X employees: food industry"	1
Cameron Connor	"Collective Co-Creation within the Open Source Software Community"	1
Meriam Miles	"What Makes Online Video Advertisements Go Viral?"	0
Dory Dean	"Social media use for corporate communications"	0

Sample Output Screen 1 b)

```
==============================
  Masters Student Details  1
==============================
Name    : Philip Morales
Title   : Working with Generation X employees: food industry
Status  : Approved
```

```
==============================
  Masters Student Details  2
==============================
Name    : Cameron Connor
Title     : Collective Co-Creation within the Open Source Software Community
Status  : Approved


==============================
  Masters Student Details  3
==============================
Name    : Meriam Miles
Title     : What Makes Online Video Advertisements Go Viral?
Status    : Pending


==============================
  Masters Student Details  4
==============================
Name    : Dory Dean
Title     : Social media use for corporate communications
Status   : Pending


~End of Details~Student~Dory Dean


~End of Details~Student~Meriam Miles


~End of Details~Student~Cameron Connor


~End of Details~Student~Philip Morales
```

Question 2

Create a class for employee details with the following specifications:

- Data members:
  - **name.** A **string** that holds the employee's name.
  - **idNumber.** An **int** variable that holds the employee's ID number.
  - **department.** A **string** that holds the name of the department where the employee works.
  - **position.** A **string** that holds the employee's job title.
- The class should have the following constructors:
  (i) A *constructor* that accepts the following values as arguments and assigns them to the appropriate member variables: employee's name, employee's ID number, department, and position.
  (ii) A *constructor* that accepts the following values as arguments and assigns them to the appropriate member variables: employee's name and ID number. The department and position fields should be assigned an empty string ("").
  (iii) A *default constructor* that assigns empty strings ("") to the *name*, *department*, and *position* member variables, and 0 to the *idNumber* member variable.
- Write appropriate *mutator* functions that store values in these member variables and *accessor* functions that return the values in these member variables.
- Once you have written the class, write a main() that creates three employee objects (one using *constructor (i)*, one using *constructor (ii) with additional mutator functions*, and one *using default constructor (iii) with mutator functions*) to hold the following data:

Name	ID	Department	Position
Susan Meyers	47899	Accounting	Vice President
Mark Jones	39119	IT	Programmer
Joy Rogers	81774	Manufacturing	Engineer

The program should store these data in three objects and then display the data for each employee on the screen. (Create additional functions to make the code more efficient, if necessary.)

Sample Output Screen

```
Name: Susan Meyers
ID Number: 47899
Department: Accounting
Position: Vice President

Name: Mark Jones
ID Number: 39119
Department: IT
Position: Programmer

Name: Joy Rogers
ID Number: 81774
Department: Manufacturing
Position: Engineer
```

Question 3

A. Given the declarations for a class ***Books*** that contains the following features:

```
#include<iostream>
#include<iomanip>
using namespace std;

class Books
{
      private:
            string isbnNo, title, author;
            float price, discountedprice, discountperc;
      public:
            Books();
            Books(string,string,string,float,float);
            void set_Data();
            void calcDiscountedPrice();
            void print();
            float getDiscountedPrice();
            ~Books();
};
```

Write the definitions of the member functions **outside of the class** based on the descriptions given below:

(a) **set_Data()**
Get user inputs for *isbnNo, title, author, price* and *discountperc.*

(b) **calcDiscountedPrice()**
Find the *discountedPrice* (the price after minus the discounted amount).

(c) **print()**
Display the book details.

(d) **getDiscountedPrice()**
Return the *discountedPrice.*

(e) **Books()**
Default constructor of the class that initialize all data members to *zero* or *empty*.

(f) **Books(string,string,string,float,float);**
Parameterized constructor of the class.

(g) **~Books()**
Display "Enjoy reading (book title)".

B. Create another function named **func(…)** that accepts an object (from Books class) by reference using a reference object as an argument. In this function, call **set_Data(), calcDiscountedPrice()** and **print()** using the reference object.

C. In main(), do the following:

(i) Create an object of class Books called **B1** assigning the following values:

ISBN : 102009912
Title : 7 Habits of Highly Effective People
Author : Stephen Covey
Original Price : RM 400.00

Discount (%) : 30

(ii) Call function **calcDiscountedPrice()** and **print()** using object **B1.**

(iii) Create an array of class Books called **B2** and size of the array is 3.

(iv) Using a *for* loop:
- call ***func(…)***, passing the object elements of array B2.
- Determine the most expensive book's price (after discount)
- Determine the number of books whose price after discount is below RM30.

(v) Display the most expensive book's price (after discount) and the number of books whose price after discount is below RM30.

Sample Output Screen

```
.....Book of the Month.....

-------------------------------------------
        Book Details
-------------------------------------------
ISBN            : 102009912
Title           : 7 Habits of Highly Effective People
Author          : Stephen Covey
Original Price  : RM 400.00
Discounted Price: RM 280.00

Now we shall enter and display data for 3 special books....

Enter ISBN          : 1892110011
Enter Title         : C++ Programming
Enter Author's name : Deitel
Enter price         : RM 42.50
Enter discount (%)  : 10

-------------------------------------------
        Book Details
-------------------------------------------
ISBN            : 1892110011
Title           : C++ Programming
Author          : Deitel
Original Price  : RM 42.50
Discounted Price: RM 38.25

Enter ISBN          : 10210211112
Enter Title         : Top 10 Influential People
Enter Author's name : Peter Stark
Enter price         : RM 30.00
Enter discount (%)  : 5

-------------------------------------------
        Book Details
-------------------------------------------
```

```
ISBN              : 10210211112
Title             : Top 10 Influential People
Author            : Peter Stark
Original Price    : RM 30.00
Discounted Price: RM 28.50

Enter ISBN              : 1201021124
Enter Title             : Python Programming Made Easy
Enter Author's name     : Julius Brevern
Enter price             : RM 120.00
Enter discount (%)      : 30

------------------------------------------------
            Book Details
------------------------------------------------
ISBN              : 1201021124
Title             : Python Programming Made Easy
Author            : Julius Brevern
Original Price    : RM 120.00
Discounted Price: RM 84.00

------------------------------------------------
The most expensive book is RM 84.00
The number of books that are below RM 30 are :1

Enjoy reading Phyton Programming Made Easy

Enjoy reading Top 10 Influential People

Enjoy reading C++ Programming

Enjoy reading 7 Habits of Highly Effective People
```