Name: Atul Kumar Reg. No: 12102801

Roll No: RD2110B79 Question No: 01

Course Code: CAP-444 SET - "H"

Q1: what do you mean by function definition inside the class and outside the class? Explain with example code.

Ans: Function of a class can be defined either inside the class or outside of the class. In both cases, the function body remain the same.

→ Inside the class: when a function is defined inside the class, the class name and the scope resolution operator are not specified in the function header.

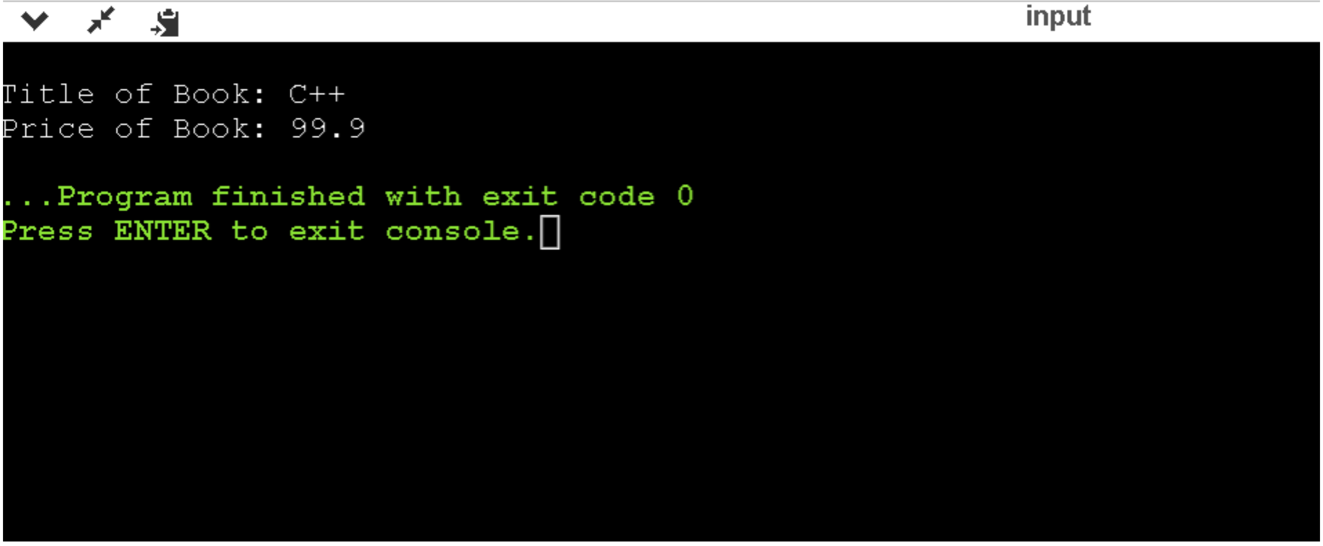
Name: Atul Kumar Reg. No: 12102801

Roll No: RD2110B79 Question No: 01

Course Code: CAP 444 SET-"H"

→ Example: Definition of a function inside a class.

```
        class book {
          string name;
          float price;
         public:
      void getdata(string s1, float price1) {
         name = s1;
         price = price1;
         }
       void display () {
        cout<<"Name of Book:"<<name;
        cout<<"Price of Book: "<<price;
       }
 };
  int main()
 {
      book b1;
      string s1 = "C++";
      b1.getdata (s1, 99.9);
      b1.display ();
      return 0;
 }
```



```
Title of Book: C++
Price of Book: 99.9

...Program finished with exit code 0
Press ENTER to exit console.
```

Name: Atul Kumar Reg. No: 12102801
Roll No: RD2110B79 Question No: 01
Course Code: CAP 444 SET – "H"

→ Outside the class: To define a function outside the class, scope resolution operator is used. Defining Big size functions outside the class decreases size of the class body which decreases execution time.

→ Example: Definition of a function outside a class.

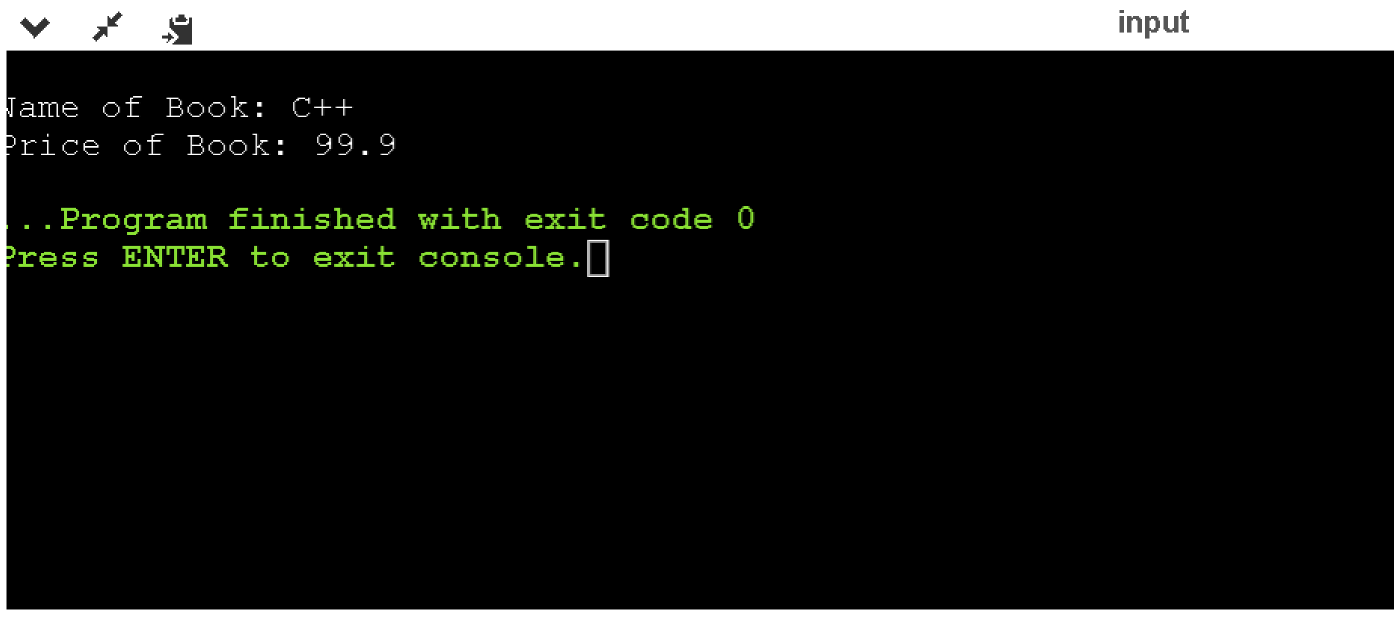
```
      class book {
        string name;
        float price;
      public:
        void getdata(string s1, float price1);
        void display();

  };
```

Name: Atul Kumar Reg. No: 12102801

Roll No: RD2110B79 Question No: 01

Course Code: CAP 444 SET-"H"

```
void book :: display() {
    cout<<"\n Name of Book : "<<name;
    cout<<"\n Price of Book : "<<price;
}
void book :: getdata(string s1, float price1)
 {
   name=s1;
   price=price1;
 }
int main()
 {
    book b1;
    string s1 = "C++";
    b1.getdata (s1, 99.9);
    b1. display();
    return 0;
 }
```



```
Name of Book: C++
Price of Book: 99.9

...Program finished with exit code 0
Press ENTER to exit console.
```

Name: Atul Kumar Reg. No: 12102801
Roll No: RD2110B79 Question No: 02
Course Code: CAP444 SET-"H"

Q2: Differentiate between parameterized and copy constructor. Illustrate the difference with the help of a program.

Ans: Parameterized constructor: It is possible to pass arguement to constructors. These arguement help initialize an object when it is created. To create parameterized constructor, simply add parameter to the way you would to any other function. When you define the constructor's body, use the parameters to initialize the Object

→ Example

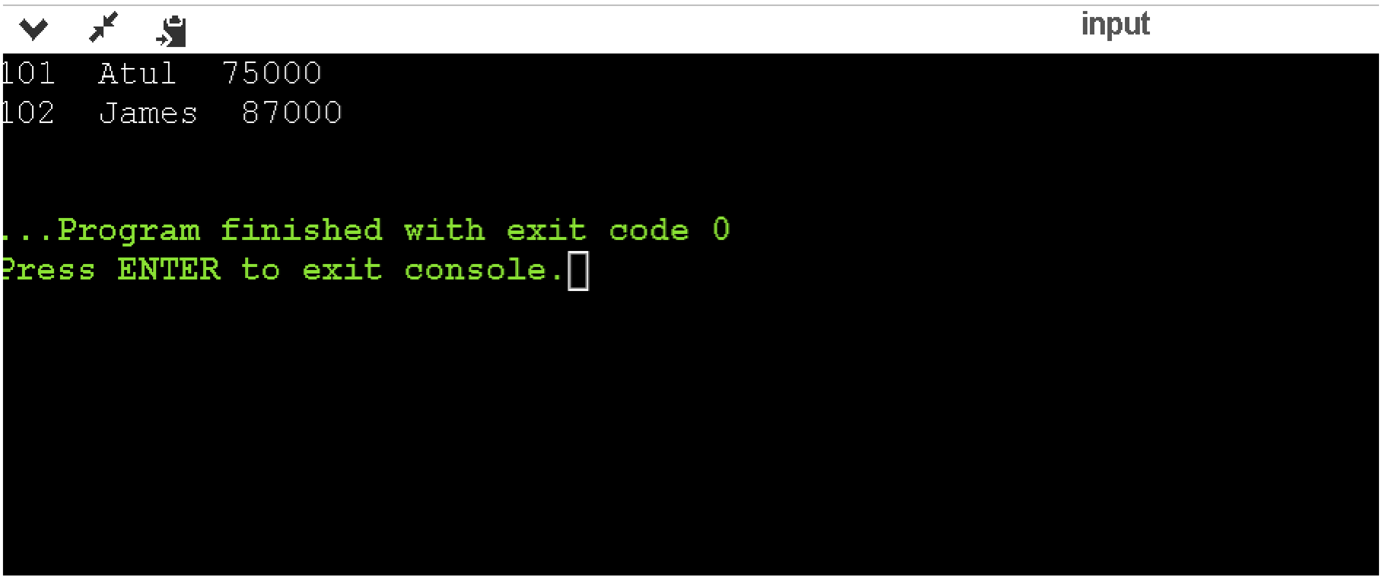
```
#include <iostream>
using namespace std;
class Emp
{
```

Name: Atul Kumar Reg. No: 12102801

Roll No: RD2110B79 Question No: 02

Course Code: CAP444 SET - "H"

```
public:
   int eid;  //data member
   string name;
   float sal;
   Emp(int i, string n, float s)//Parameterized construct.
    {
      eid = i;
      name = n;
      sal = s;
    }
  void display()
    {
    cout<<eid<<"  "<<name<<"  "<<sal
                              << endl;
    }
int main(void) {
 Emp e1 = Emp(101, "Atul", 75000);
 Emp e2 = Emp(102, "James", 87000);
  e1.display();
  e2.display();
  return 0;
}
```



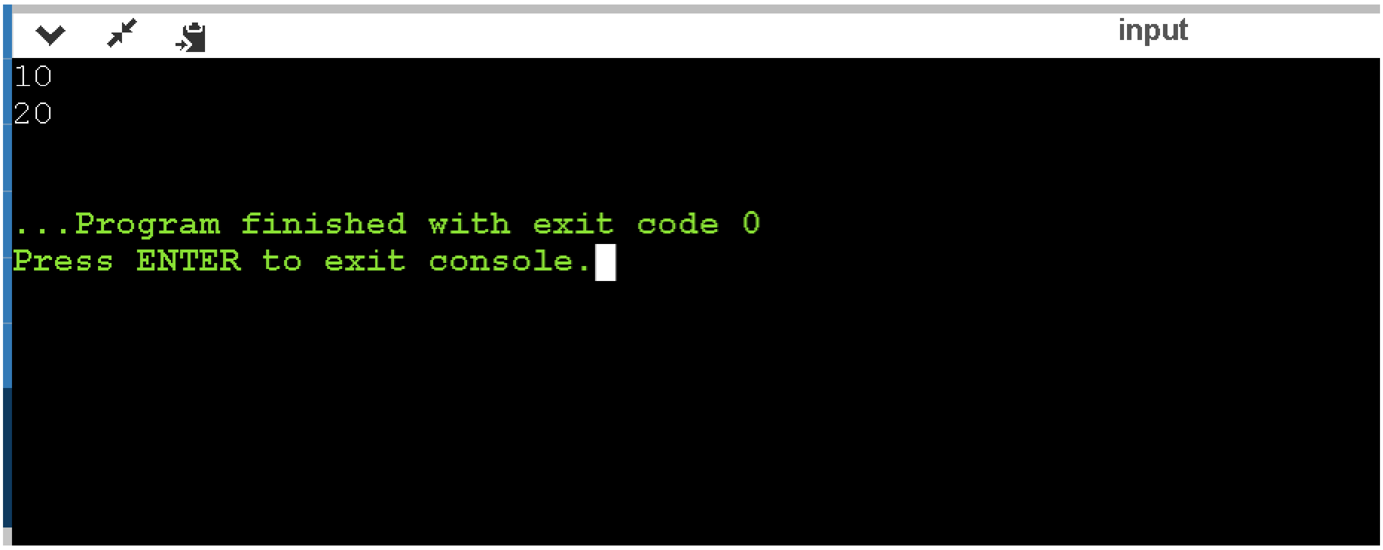
```
101   Atul   75000
102   James   87000


...Program finished with exit code 0
Press ENTER to exit console.
```

Name: Atul Kumar Reg. No: 12102801
Roll No: RD2110B79 Question No: 02
Course Code: CAP444 SET-"H"

Copy Constructor: A copy constructor is a member function that initializes an object using another object of the same class. The copy constructor is used to copy an object to pass it as an argument to a function and copy an object to return it from a function.

→ Example

```
#include <iostream>
using namespace std;
class point
 {
 int x, y;
 public:
 point(int a, int b)
  {
  x=a;
  y=b;
  }
point (const point &z) {
 x = z.x;
 y = z.y;
 }
int getX() {
 return x;
}
int getY() {
return y;
}
};
 int main()
{
point p1(10,20);
point p2= P1;
cout<< p1.getX()<<endl;
cout<<p1.getY()<<endl;
    return 0;
}
```



```
10
20


...Program finished with exit code 0
Press ENTER to exit console.
```

Name: Atul Kumar Reg. No: 12102801

Roll No: RD2110B79 Question No: 03

Course Code: CAP444 SET-"H"

Q3: Create a function to find only the root value of x in any quadratic equation ax^2+bx+c. The function will take three arguments a as the coefficient of x^2 b as the coefficient of x c as the constant term Implement it using friend function.

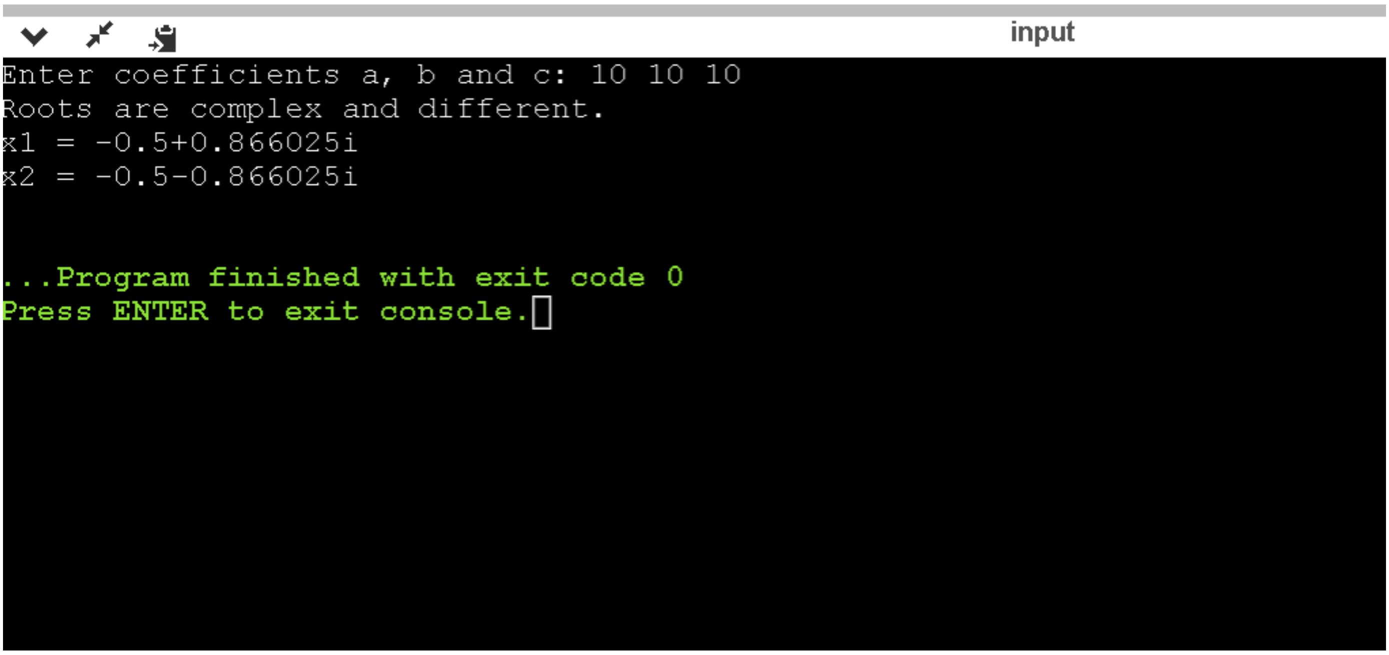
Sol^n:
```
#include <iostream>
#include <cmath>
using namespace std;
class Quadratic {
   float a, b, c, x1, x2, discriminant, realPart,
                        imaginaryPart;
   public:
   friend void root(Quadratic);
   void takeInput(){
     cout<<"Enter coefficients a, b and c: ";
     cin>> a >> b >> c;
     discriminant = b*b-4*a*c;
   }
};
```

Name: Atul Kumar Reg. No: 12102801

Roll No: RD2110B79 Question No: 03

Course Code: CAP444 SET-"H"

```
void root(Quadratic q1){
  if (q1.discriminant>0){
  q1.x1=(-q1.b+sqrt(q1.discriminant))/(2*q1.a);
  q1.x2=(-q1.b-sqrt(q1.discriminant))/(2*q1.a);
 cout<<"Roots are real and different."<<endl;
  cout<<"x1="<<q1.x1 <<endl;
  cout<<"x2="<<q1.x2<<endl;
 }
else if (q1.discriminant==0) {
  cout<<"Roots are real and same."<<endl;
   q1.x1=-q1.b/(2*q1.a);
  cout<<"x1=x2="<<q1.x1<<endl;
}
else
 {
   q1.realPart=-q1.b/(2*q1.a);
  q1.imaginaryPart=sqrt(-q1.discriminant)/(2*q1.a);
 cout<<"Roots ase complex and different."<<endl;
 cout<<"x1="<<q1.realPart<<"+"<<q1.imag
               inaryPart<<"i"<<endl;
cout<<"x2= "<<q1.realPart<<"-"
          <<q1.imaginaryPart <<"i"<<endl;
 }
}
int main()
 {
    Quadratic q1;
    q1.takeInput();
    root(q1);
    return 0;
 }
```



```
Enter coefficients a, b and c: 10 10 10
Roots are complex and different.
x1 = -0.5+0.866025i
x2 = -0.5-0.866025i


...Program finished with exit code 0
Press ENTER to exit console.
```