Bangladesh University of Business and Technology

(BUBT)

Lab Report

Lab Report No: 03

Course: Object Oriented Programming Lab

Course Code:CSE 112

Date of Submission:

Submitted By:
Name: Argho Ghosh
ID: 20254103276
Intake: 56
Section: 05
Program: B.Sc. in CSE

Bangladesh University of Business and Technology (BUBT)

Submitted To:
Iffat Ara Sanzida
Lecturer
Department of Computer Science & Engineering

Bangladesh University of Business and Technology (BUBT)

Teacher's Signature

Task 1:

Design a class named Product to represent a product in a store.
The class should include private attributes for the product's name (productName), ID (productID), and price (price).
Implement a default constructor that initializes productName to "Unnamed",
productID to 0, and price to 0.0, and a parameterized constructor that allows setting these attributes to specific values upon object creation.
Use a method to display the product information (displayDetails).
The class should also have a destructor that outputs a message when an object is destroyed.
In the main function, create objects using both constructors, modify the default object's details, and display the information for each product.

Code:

```
#include<bits/stdc++.h>
using namespace std;
class product
{
private :
   string pName;long long pID;int pPrice;
public:
product()
{
   pName = "Unnamed";
   pID = 0;
   pPrice = 0;
}
product(string a,long long a1, int a2)
{
   pName = a;
   pID = a1;
   pPrice = a2;
}
~product()
{
   cout<<"Thank you!"<<endl;
}
void display()
{
```

```
    cout<<"Product Name = "<<pName<<endl;
    cout<<"Product ID = "<<pID<<endl;
    cout<<"Product price = "<<pPrice<<endl;
}
};
int main()
{
    product p;
    p.display();
    product p1("Cake",22556255,25);
    p1.display();


}
```

Output:

Task 2:

Develop a C++ class named Point that represents a point in a 2D coordinate system, with two private integer variables x and y to store the coordinates. The class should include a constructor that initializes these coordinates upon object creation and a copy constructor that allows one Point object to be initialized as a copy of another. Additionally, the class should provide getter functions to access the x and y values. Create a distance function to calculate euclidean distance between two objects.

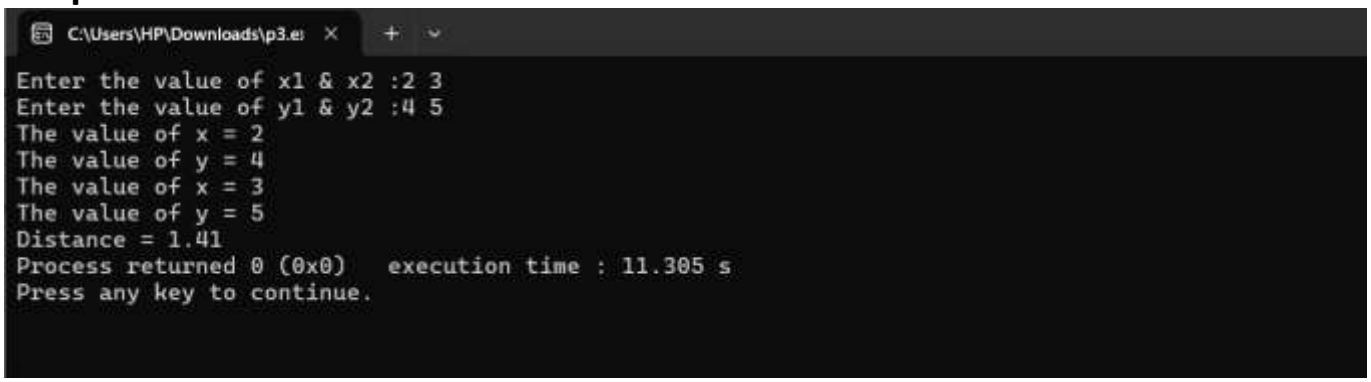
Code:

```
#include<bits/stdc++.h>
using namespace std;
class point
{
private:
    int x,y;
```

```
public:

   point(int a,int b)
   {
      x = a;
      y = b;
   }
   point(point &p)
   {
      x = p.x;
      y = p.y;
   }

   int getX()
   {
      return x;

   }
   int getY()
   {
      return y;
   }

   void display()
   {
      cout<<"The value of x = "<<x<<endl;
      cout<<"The value of y = "<<y<<endl;
   }


};
void distance(point p1,point p2)
{
   int v1 = p1.getX() - p2.getX();
   v1 = pow(v1,2);
   int v2 = p1.getY() - p2.getY();
   v2 = pow(v2, 2);

   double d = sqrt((double)v1+v2);
   cout<<setprecision(3)<<"Distance = "<<d;
```

```
}

int main()
{
   int x1,x2,y1,y2;
   cout<<"Enter the value of x1 & x2 :";
   cin>>x1>>x2;
   cout<<"Enter the value of y1 & y2 :";
   cin>>y1>>y2;
   point p1(x1,y1);
   p1.display();
   point p2(x2,y2);
   point p3(p2);
   p3.display();
   distance(p1,p2);
}
```

Output:



Task 3:

Design a class named Book to encapsulate information about a book such as: title, author, and publicationYear. Implement the following:

- *A non-parameterized (default) constructor that initializes title and author to "Unknown" and publicationYear to 0.*
- *A destructor that displays a message indicating which Book object (by title) is being destroyed.*
- *Public member functions to set the book details by taking input from the user and to display the book's information. Inside the setter function for the*

publication year, implement validation logic to ensure the year is not in the future (i.e., it should not exceed the current year). If the input is invalid, set the year to 0 and display a warning message.

- *In the main() function, create an object of the Book class, prompt the user to input book details using the setter functions, display the entered information, then create additional Book objects using the default constructor and display*

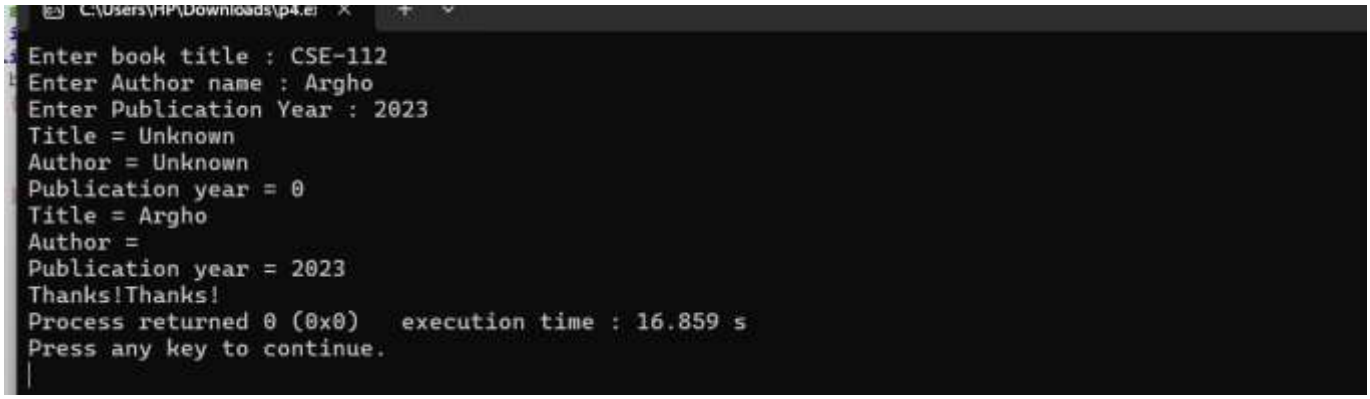
their details.

Code:

```
#include<bits/stdc++.h>
using namespace std;
class book
{
private:
   string title,author;
   int pYear;
public:
   book()
   {
      title = "Unknown";
      author = "Unknown";
      pYear = 0;
   }
//    string getTitle()
//    {
//       return title;
//    }
//    string getAuthor()
//    {
//       return author;
//    }
//    int getPublicYear()
//    {
//       return pYear;
//    }
   book(string s1, string s2, int p)
   {
      title = s1;
      author = s2;
      pYear = p;
   }
   void display()
   {
```

```
        cout<<"Title = "<<title<<endl;
        cout<<"Author = "<<author<< endl;
        cout<<"Publication year = "<<pYear <<endl;
    }
    ~book()
    {
        cout<<"Thanks!";
    }

};
int main()
{
    string a1,a2;
    int p;
    cout<<"Enter book title : ";
    getline(cin,a1);
    cout<<"Enter Author name : ";
    getline(cin,a1);
    cout<<"Enter Publication Year : ";
    cin>>p;

    if(p>=2026)
    {
        cout<<"Invalid input"<<endl;
        p=0;
    }
    book b1;
    b1.display();
    book b(a1,a2,p);

    b.display();

}
```

Output:


C:\Users\HP\Downloads\p4.ex
Enter book title : CSE-112
Enter Author name : Argho
Enter Publication Year : 2023
Title = Unknown
Author = Unknown
Publication year = 0
Title = Argho
Author =
Publication year = 2023
Thanks!Thanks!
Process returned 0 (0x0)   execution time : 16.859 s
Press any key to continue.