OOP Assignment

“Inheritance”

Submitted by: Muhammad Faisal Kamran

Submitted to: Sir Javed

Registration Number: 2023-BS-AI-025

Department: BS-AI

Section: A

Date: 2/5/2024

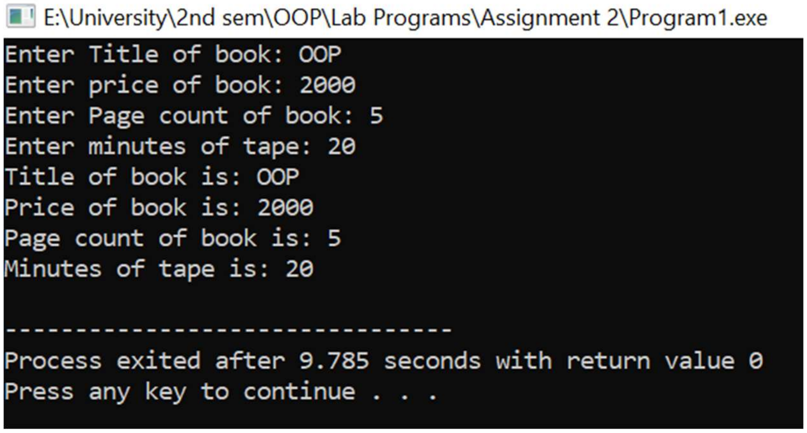
Problem 1

Source Code

```
// File: Program1.cpp

// Date: 22-05-2024

// Name: Muhammad Faisal Kamran

// Registration No: 2023-BS-AI-025

/* Imagine a publishing company that markets both book and audiocasts versions of its works.
Create a class publication that stores the title (a string) and price (type float) of a publication.
From this class derive two classes: book, which adds a page count (type int), and tape, which
adds a playing time in minutes (type float). Each of these three classes should have a getdata()
function to get its data from the user at the keyboard, and a putdata() function to display its data.
Write a main() program to test the book and tape classes by creating instances of them, asking
the user to fill in data with getdata(), and then displaying the data with putdata().*/

#include<iostream>

using namespace std;

class publication
```

```
{
private:
string title;
float price;
public:
        void getpublicationdata(){
                cout<<"Enter Title of book: ";
                cin>>title;
                cout<<"Enter price of book: ";
                cin>>price;
        }
        void putpublicationdata(){
                cout<<"Title of book is: "<<title<<endl;
                cout<<"Price of book is: "<<price<<endl;
        }
};
class book:public publication
{
private:
int page_count;
public:
        void getbookdata(){
                cout<<"Enter Page count of book: ";
                cin>>page_count;
        }
        void putbookdata(){
                cout<<"Page count of book is: "<<page_count<<endl;
        }
};
class tape:public publication
{
private:
float minutes;
public:
        void gettapedata(){
                cout<<"Enter minutes of tape: ";
                cin>>minutes;
        }
        void puttapedata(){
                cout<<"Minutes of tape is: "<<minutes<<endl;
        }       };
int main(){
book faisal;
tape danish;
faisal.getpublicationdata();
faisal.getbookdata();
danish.gettapedata();
faisal.putpublicationdata();
faisal.putbookdata();
danish.puttapedata();}
```

Output



E:\University\2nd sem\OOP\Lab Programs\Assignment 2\Program1.exe

```
Enter Title of book: OOP
Enter price of book: 2000
Enter Page count of book: 5
Enter minutes of tape: 20
Title of book is: OOP
Price of book is: 2000
Page count of book is: 5
Minutes of tape is: 20

--------------------------------
Process exited after 9.785 seconds with return value 0
Press any key to continue . . .
```

Problem 2

Source Code

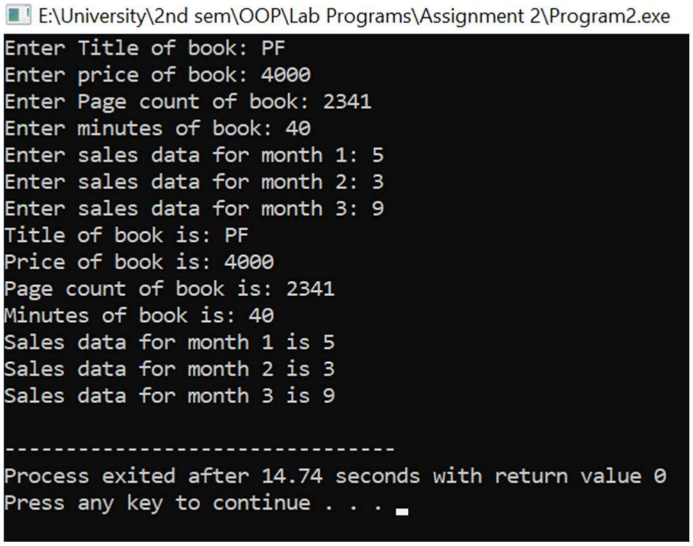
```
// File: Program2.cpp

// Date: 22-05-2024

// Name: Muhammad Faisal Kamran

// Registration No: 2023-BS-AI-025

/* Start with the publication, book, and tape classes of Question 1. Add a base class sales that
holds an array of three floats so that it can record the dollar sales of a particular publication for
the last three months. Include a getdata() function to get three sales amounts from the user, and a
putdata() function to display the sales figures. Alter the book and tape classes so they are derived
from both publication and sales. An object of class book or tape should input and output sales
data along with its other data. Write a main() function to create a book object and a tape object
and exercise their input/output capabilities. */


#include<iostream>

using namespace std;

class publication

{

private:

string title;

float price;

public:

	void getpublicationdata(){

		cout<<"Enter Title of book: ";

		cin>>title;

		cout<<"Enter price of book: ";
```

```
        cin>>price;
    }
    void putpublicationdata(){
        cout<<"Title of book is: "<<title<<endl;
        cout<<"Price of book is: "<<price<<endl;
    }
};
class sales{
private:
float arr[3];
public:
void getsalesdata(){

    for(int i=0;i<3;i++){
    cout<<"Enter sales data for month "<<i+1<<": ";
        cin>>arr[i];
    }
}
void putsalesdata(){
    for(int i=0;i<3;i++){
        cout<<"Sales data for month "<<i+1<<" is ";
        cout<<arr[i]<<endl;
    }
}
};
```

```
class book:public publication, public sales
{
private:
int page_count;
public:
        void getbookdata(){
                cout<<"Enter Page count of book: ";
                cin>>page_count;
        }
        void putbookdata(){
                cout<<"Page count of book is: "<<page_count<<endl;
        }
};
class tape:public publication,public sales
{
private:
float minutes;
public:
        void gettapedata(){
                cout<<"Enter minutes of book: ";
                cin>>minutes;
        }
        void puttapedata(){
                cout<<"Minutes of book is: "<<minutes<<endl;}
};
```

```
int main(){

book faisal;

tape danish;

faisal.getpublicationdata();

faisal.getbookdata();

danish.gettapedata();

danish.getsalesdata();

faisal.putpublicationdata();

faisal.putbookdata();

danish.puttapedata();

danish.putsalesdata();

}
```

Output


E:\University\2nd sem\OOP\Lab Programs\Assignment 2\Program2.exe

```
Enter Title of book: PF
Enter price of book: 4000
Enter Page count of book: 2341
Enter minutes of book: 40
Enter sales data for month 1: 5
Enter sales data for month 2: 3
Enter sales data for month 3: 9
Title of book is: PF
Price of book is: 4000
Page count of book is: 2341
Minutes of book is: 40
Sales data for month 1 is 5
Sales data for month 2 is 3
Sales data for month 3 is 9

--------------------------------
Process exited after 14.74 seconds with return value 0
Press any key to continue . . .
```

Problem 3

Source Code

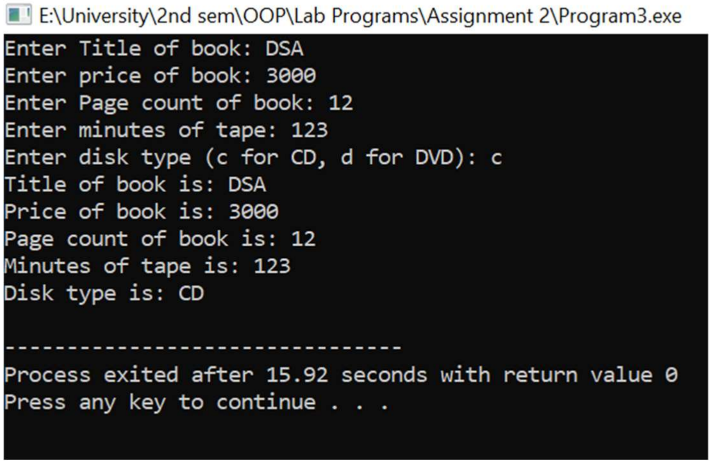
```
// File: Program3.cpp

// Date: 22-05-2024

// Name: Muhammad Faisal Kamran

// Registration No: 2023-BS-AI-025

/*Assume that the publisher in Queston 1 and 3 decides to add a third way to distribute books: on
computer disk, for those who like to do their reading on their laptop. Add a disk class that, like
book and tape, is derived from publication. The disk class should incorporate the same member
functions as the other classes. The data item unique to this class is the disk type: either CD or
DVD. You can use an enum type to store this item. The user could select the appropriate type by
typing c or d. */

#include <iostream>

using namespace std;

class publication {
```

```
private:
    string title;
    float price;
public:
    void getpublicationdata() {
        cout << "Enter Title of book: ";
        cin >> title;
        cout << "Enter price of book: ";
        cin >> price;
    }

    void putpublicationdata() {
        cout << "Title of book is: " << title << endl;
        cout << "Price of book is: " << price << endl;
    }
};

class book : public publication {
private:
    int page_count;
public:
    void getbookdata() {
        cout << "Enter Page count of book: ";
        cin >> page_count;
    }
    void putbookdata() {
        cout << "Page count of book is: " << page_count << endl;
    }
};

class tape : public publication {
private:
    float minutes;
public:
    void gettapedata() {
        cout << "Enter minutes of tape: ";
        cin >> minutes;
    }

    void puttapedata() {
        cout << "Minutes of tape is: " << minutes << endl;
    }
};

enum DiskType { CD, DVD };

class disk : public publication {
private:
    DiskType disk_type;
```

```
public:
    void getdiskdata() {
        char type;
        cout << "Enter disk type (c for CD, d for DVD): ";
        cin >> type;
        if (type == 'c' || type == 'C') {
            disk_type = CD;
        } else if (type == 'd' || type == 'D') {
            disk_type = DVD;
        } else {
            cout << "Invalid disk type. Defaulting to CD." << endl;
            disk_type = CD;
        }
    }

    void putdiskdata() {
        cout << "Disk type is: " << (disk_type == CD ? "CD" : "DVD") << endl;
    }
};
int main() {
    book faisal;
    tape danish;
    disk kamran;
faisal.getpublicationdata();
    faisal.getbookdata();
```

```
   danish.gettapedata();

   kamran.getdiskdata();

   faisal.putpublicationdata();

   faisal.putbookdata();

   danish.puttapedata();

   kamran.putdiskdata();

   return 0;

}
```

Output



E:\University\2nd sem\OOP\Lab Programs\Assignment 2\Program3.exe

```
Enter Title of book: DSA
Enter price of book: 3000
Enter Page count of book: 12
Enter minutes of tape: 123
Enter disk type (c for CD, d for DVD): c
Title of book is: DSA
Price of book is: 3000
Page count of book is: 12
Minutes of tape is: 123
Disk type is: CD

--------------------------------
Process exited after 15.92 seconds with return value 0
Press any key to continue . . .
```

Problem 4

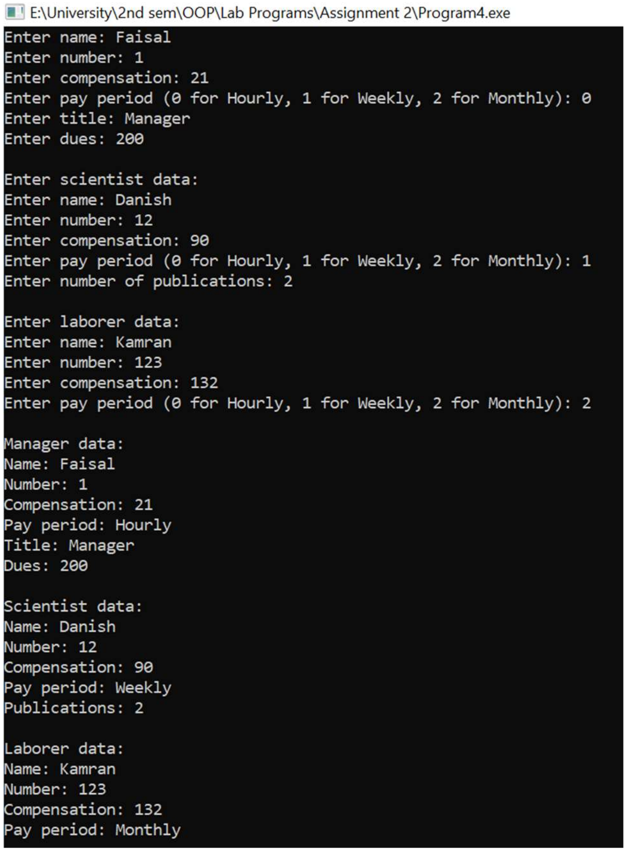
Source Code

```
// File: Program4.cpp

// Date: 22-05-2024

// Name: Muhammad Faisal Kamran

// Registration No: 2023-BS-AI-025

/*Derive a class called employee2 from the employee class in the EMPLOY program in this
chapter. This new class should add a type double data item called compensation, and also an
enum type called period to indicate whether the employee is paid hourly, weekly, or monthly. For
simplicity you can change the manager, scientist, and laborer classes so they are derived from
employee2 instead of employee. However, note that in many circumstances it might be more in
the spirit of OOP to create a separate base class called compensation and three new classes
manager2, scientist2, and laborer2, and use multiple inheritance to derive these three classes
from the original manager, scientist, and laborer classes and from compensation. This way none
of the original classes needs to be modified

*/

#include <iostream>

#include <string>

using namespace std;
```

```
enum Period { HOURLY, WEEKLY, MONTHLY };

class Employee {

protected:

    string name;

    unsigned long number;

public:

    void getdata() {

        cout << "Enter name: ";

        cin >> name;

        cout << "Enter number: ";

        cin >> number;

    }

    void putdata() const {

        cout << "Name: " << name << "\n";

        cout << "Number: " << number << "\n";

    }

};

class Employee2 : public Employee {

private:

    double compensation;

    Period period;

public:

    void getdata() {

        Employee::getdata();

        cout << "Enter compensation: ";
        cin >> compensation;
        int periodInput;
        cout << "Enter pay period (0 for Hourly, 1 for Weekly, 2 for Monthly): ";
        cin >> periodInput;
        period = static_cast<Period>(periodInput);
    }
    void putdata() const {
        Employee::putdata();
        cout << "Compensation: " << compensation << "\n";
        cout << "Pay period: ";
        switch (period) {
            case HOURLY: cout << "Hourly\n"; break;
            case WEEKLY: cout << "Weekly\n"; break;
            case MONTHLY: cout << "Monthly\n"; break;
        }
    }
};

class Manager2 : public Employee2 {
private:
    string title;
    double dues;
public:
    void getdata() {
        Employee2::getdata();
        cout << "Enter title: ";
        cin >> title;
        cout << "Enter dues: ";
        cin >> dues;
    }
    void putdata() const {
        Employee2::putdata();
        cout << "Title: " << title << "\n";
        cout << "Dues: " << dues << "\n";
    }
};

class Scientist2 : public Employee2 {
private:
    int publications;
public:
    void getdata() {
        Employee2::getdata();
        cout << "Enter number of publications: ";
        cin >> publications;
    }
    void putdata() const {
        Employee2::putdata();
        cout << "Publications: " << publications << "\n";
    }
```

```
};


class Laborer2 : public Employee2 {
    // No additional data members
};


int main() {
    Manager2 mgr;
    Scientist2 sci;
    Laborer2 lab;
    cout << "Enter manager data:\n";
    mgr.getdata();
    cout << "\nEnter scientist data:\n";
    sci.getdata();
    cout << "\nEnter laborer data:\n";
    lab.getdata();
    cout << "\nManager data:\n";
    mgr.putdata();
    cout << "\nScientist data:\n";
    sci.putdata();
    cout << "\nLaborer data:\n";
    lab.putdata();


    return 0;
}
```

Output



E:\University\2nd sem\OOP\Lab Programs\Assignment 2\Program4.exe

```
Enter name: Faisal
Enter number: 1
Enter compensation: 21
Enter pay period (0 for Hourly, 1 for Weekly, 2 for Monthly): 0
Enter title: Manager
Enter dues: 200

Enter scientist data:
Enter name: Danish
Enter number: 12
Enter compensation: 90
Enter pay period (0 for Hourly, 1 for Weekly, 2 for Monthly): 1
Enter number of publications: 2

Enter laborer data:
Enter name: Kamran
Enter number: 123
Enter compensation: 132
Enter pay period (0 for Hourly, 1 for Weekly, 2 for Monthly): 2

Manager data:
Name: Faisal
Number: 1
Compensation: 21
Pay period: Hourly
Title: Manager
Dues: 200

Scientist data:
Name: Danish
Number: 12
Compensation: 90
Pay period: Weekly
Publications: 2

Laborer data:
Name: Kamran
Number: 123
Compensation: 132
Pay period: Monthly
```

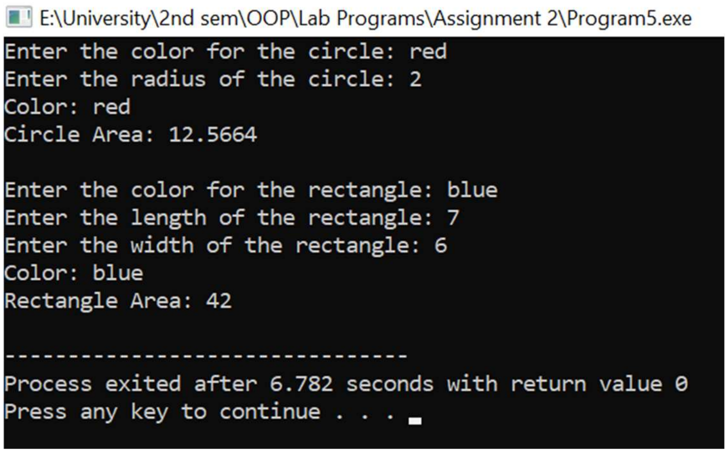
Problem 5

Source Code

```
// File: Program5.cpp

// Date: 22-05-2024

// Name: Muhammad Faisal Kamran

// Registration No: 2023-BS-AI-025

/*Create a simple inheritance hierarchy for a Shape class, Circle class, and Rectangle class. The
Shape class should be the base class, and Circle and Rectangle should be derived classes.
Implement the following in C++

Shape Class:

Attributes: color (type std::string).

Methods: A constructor to initialize the color and a method printColor to display the color.

Circle Class:

Attributes: radius (type double).

Methods: A constructor to initialize the color and radius, a method calculateArea to calculate the
area of

the circle (area = p * radius * radius), and a method printArea to display the area.

Rectangle Class:

Attributes: length and width (type double).

Methods: A constructor to initialize the color, length, and width, a method calculateArea to
calculate the

area of the rectangle (area = length * width), and a method printArea to display the area.

*/

#include <iostream>

#include <cmath>

using namespace std;
```

```
class Shape {
protected:
    string color;
public:
    Shape(const string& color) : color(color) {}

    void printColor() const {
        cout << "Color: " << color << endl;
    }
};
class Circle : public Shape {
private:
    double radius;
public:
    Circle(const string& color, double radius) : Shape(color), radius(radius) {}
    double calculateArea() const {
        return M_PI * radius * radius;
    }
    void printArea() const {
        cout << "Circle Area: " << calculateArea() << endl;
    }
};
class Rectangle : public Shape {
private:
    double length;
    double width;
public:
    Rectangle(const string& color, double length, double width) : Shape(color), length(length), width(width) {}
    double calculateArea() const {
        return length * width;
    }
    void printArea() const {
        cout << "Rectangle Area: " << calculateArea() << endl;
    }
};
int main() {
    string color;
    double radius, length, width;
    cout << "Enter the color for the circle: ";
    cin >> color;
    cout << "Enter the radius of the circle: ";
    cin >> radius;
    Circle circle(color, radius);
    circle.printColor();
    circle.printArea();
    cout << "\nEnter the color for the rectangle: ";
    cin >> color;
    cout << "Enter the length of the rectangle: ";
    cin >> length;
```

Output



E:\University\2nd sem\OOP\Lab Programs\Assignment 2\Program5.exe

```
Enter the color for the circle: red
Enter the radius of the circle: 2
Color: red
Circle Area: 12.5664

Enter the color for the rectangle: blue
Enter the length of the rectangle: 7
Enter the width of the rectangle: 6
Color: blue
Rectangle Area: 42

--------------------------------
Process exited after 6.782 seconds with return value 0
Press any key to continue . . .
```

```
    cout << "Enter the width of the rectangle: ";
    cin >> width;
    Rectangle rectangle(color, length, width);
    rectangle.printColor();
    rectangle.printArea();
    return 0;
}
```

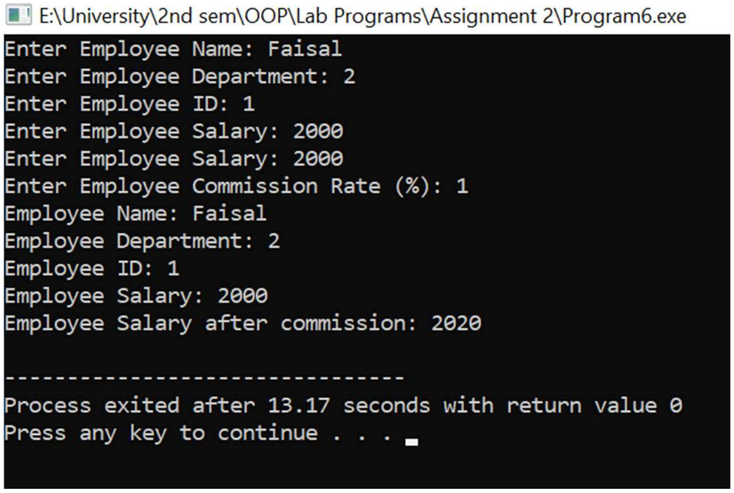
Problem 6

Source Code

```
// File: Program6.cpp
// Date: 22-05-2024
// Name: Muhammad Faisal Kamran
// Registration No: 2023-BS-AI-025
/*Design a class hierarchy for an Employee management system. The base class should be
Employee with derived classes SalariedEmployee and CommissionEmployee. Each class should
have appropriate data members and member functions to handle the specific attributes and
behaviors of each type of employee.

Employee: Should have data members for name, employee ID, and department. It should also
have member functions to get and set these values.

Salaried Employee: Inherits from Employee and adds a data member for annual Salary. It should
have member functions to get and set the salary, and to calculate the monthly pay.

Commission Employee: Inherits from Employee and adds data members for sales and
commission Rate. It should have member functions to get and set these values, and to calculate
the total pay based on sales and commission rate
*/
#include<iostream>
using namespace std;
```

```
class Employee {
private:
    string name, department;
    int EmpID;

public:
    void getEmpdata() {
        cout << "Enter Employee Name: ";
        cin >> name;
        cout << "Enter Employee Department: ";
        cin >> department;
        cout << "Enter Employee ID: ";
        cin >> EmpID;
    }

    void putEmpdata() {
        cout << "Employee Name: " << name << endl;
        cout << "Employee Department: " << department << endl;
        cout << "Employee ID: " << EmpID << endl;
    }
};

class SalariedEmployee : public Employee {
private:
    int salary;

public:
    void getSEmpdata() {
        cout << "Enter Employee Salary: ";
        cin >> salary;
    }

    void putSEmpdata() {
        cout << "Employee Salary: " << salary << endl;
    }
};

class CommissionedEmployee : public Employee {
private:
    int salary;
    float commissionrate;

public:
    void getCEmpdata() {
        cout << "Enter Employee Salary: ";
        cin >> salary;
        cout << "Enter Employee Commission Rate (%): ";
        cin >> commissionrate;
    }
    void putCEmpdata() {
        float commission = (salary * commissionrate) / 100;
        float totalSalary = salary + commission;
        cout << "Employee Salary after commission: " << totalSalary << endl;
    }
};

int main() {
    SalariedEmployee faisal;
    CommissionedEmployee danish;

    faisal.getEmpdata();
    faisal.getSEmpdata();
    danish.getCEmpdata();

    faisal.putEmpdata();
    faisal.putSEmpdata();
    danish.putCEmpdata();

    return 0;
}
```

Output



E:\University\2nd sem\OOP\Lab Programs\Assignment 2\Program6.exe

```
Enter Employee Name: Faisal
Enter Employee Department: 2
Enter Employee ID: 1
Enter Employee Salary: 2000
Enter Employee Salary: 2000
Enter Employee Commission Rate (%): 1
Employee Name: Faisal
Employee Department: 2
Employee ID: 1
Employee Salary: 2000
Employee Salary after commission: 2020

--------------------------------
Process exited after 13.17 seconds with return value 0
Press any key to continue . . .
```