LAB TASK:1

3873-FBAS/BSCS/F18-B

Course Code: CS211

Course Name: OOP

Submitted To: Mam Parkha

Submitted By: Zainab Israr

QNO:1

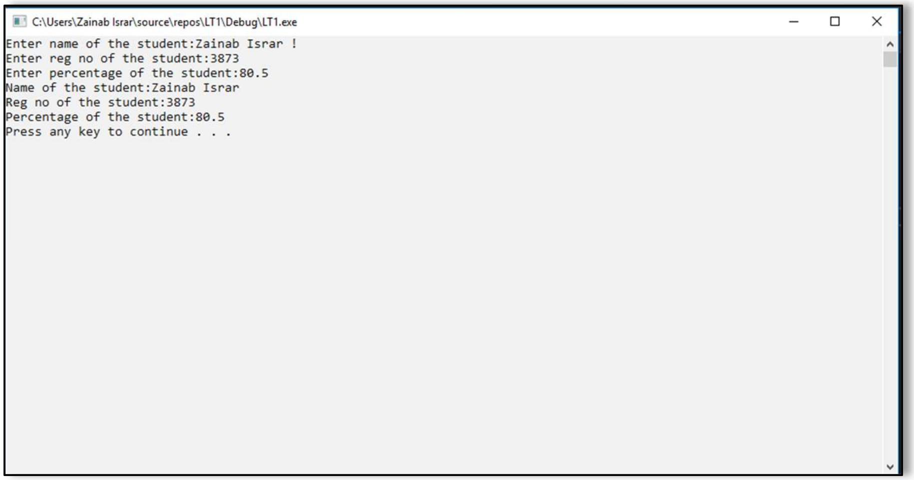
“Student.h”

```
#pragma once
#include<iostream>
#include<string>
using namespace std;
class Student
{
        string Name;
        int regNo;
        float percentage;
public:
        void getData()
        {
                cout << "Enter name of the student:";
                getline(cin, Name,'!');
                cout << "Enter reg no of the student:";
                cin >> regNo;
                cout << "Enter percentage of the student:";
                cin >> percentage;
        }
        void displayData()
        {
                cout <<"Name of the student:"<<Name<<endl
                    << "Reg no of the student:"<<regNo<<endl
                    << "Percentage of the student:"<<percentage<<endl;
        }
};
```

“Student.cpp”

```
#include "Student.h"
#include<iostream>
#include<string>
using namespace std;
int main()
{
        Student obj;
        obj.getData();
        obj.displayData();
        system("pause");
        return 0;
}
```

OUTPUT:



```
Enter name of the student:Zainab Israr !
Enter reg no of the student:3873
Enter percentage of the student:80.5
Name of the student:Zainab Israr
Reg no of the student:3873
Percentage of the student:80.5
Press any key to continue . . .
```

QNO:2

<u>“Account.h”</u>

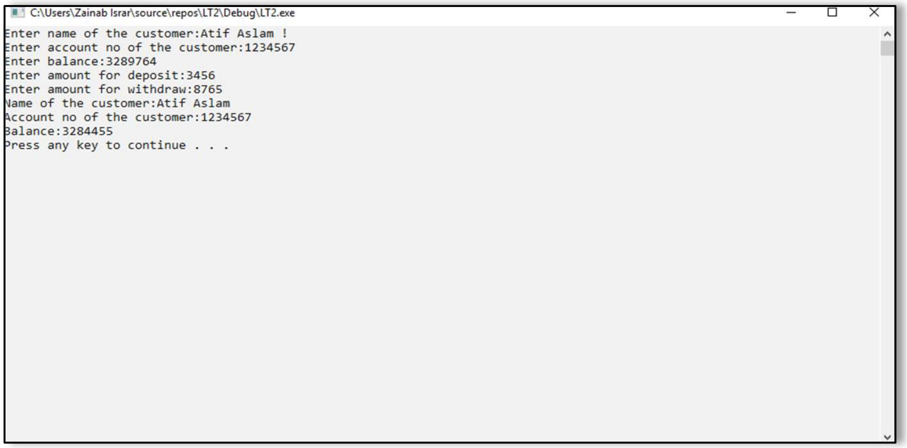
```
#pragma once
#include<iostream>
#include<string>
using namespace std;
class Account
{
    int AccountNO, Balance;
    string Name;
public:
    void getData()
    {
        cout << "Enter name of the customer:";
        getline(cin, Name, '!');
        cout << "Enter account no of the customer:";
        cin >>AccountNO;
        cout << "Enter balance:";
        cin >> Balance;
    }
    void displayData()
    {
        cout << "Name of the customer:" << Name << endl
            << "Account no of the customer:" << AccountNO << endl
            << "Balance:" <<Balance<< endl;
    }
    void withDraw()
```

```
        {
                int wd;
                cout << "Enter amount for withdraw:";
                cin >> wd;
                Balance -= wd;
        }
        void Deposit()
        {
                int dp;
                cout << "Enter amount for deposit:";
                cin >> dp;
                Balance += dp;
        }
};
```

<u>"Account.cpp"</u>

```
#include "Account.h"
#include<iostream>
#include<string>
using namespace std;
int main()
{
        Account obj;
        obj.getData();
        obj.Deposit();
        obj.withDraw();
        obj.displayData();
        system("pause");
        return 0;
}
```

OUTPUT:



```
Enter name of the customer:Atif Aslam !
Enter account no of the customer:1234567
Enter balance:3289764
Enter amount for deposit:3456
Enter amount for withdraw:8765
Name of the customer:Atif Aslam
Account no of the customer:1234567
Balance:3284455
Press any key to continue . . .
```

QNO:3

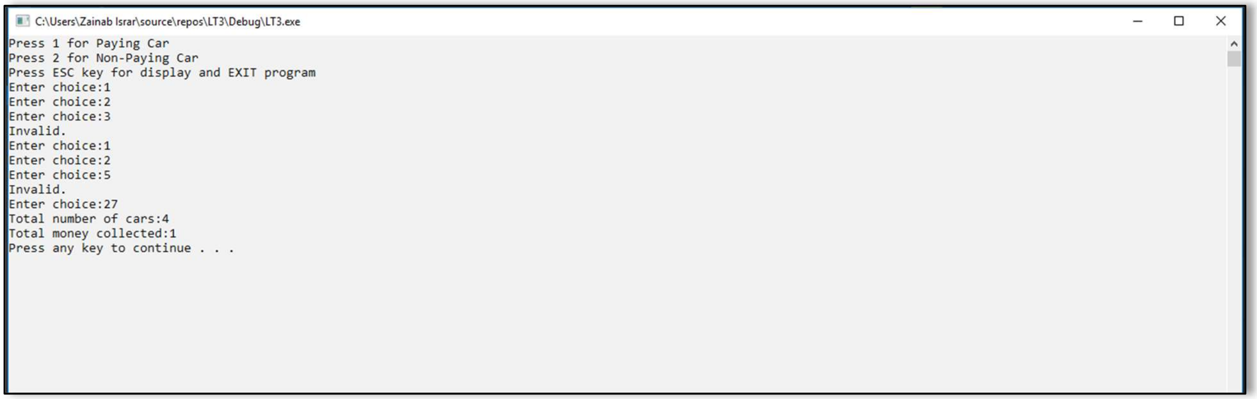
“TollBooth.h”

```
#pragma once
#include<iostream>
using namespace std;
class TollBooth
{
        unsigned int NoOfCars;
        double Amount;
public:
        TollBooth()
        {
                NoOfCars = 0;
                Amount = 0;
        }
        void payingCar()
        {
                NoOfCars++;
                Amount += 0.50;
        }
        void NonPaying()
        {
                NoOfCars++;
        }
        void Display()
        {
                cout << "Total number of cars:" << NoOfCars << endl
                        << "Total money collected:" << Amount << endl;
        }
};
```

“TollBooth.cpp”

```
#include "TollBooth.h"
#include<iostream>
#define ESC 27
using namespace std;
int main()
{
        TollBooth obj;
        cout << "Press 1 for Paying Car" << endl
                << "Press 2 for Non-Paying Car" << endl
                << "Press ESC key for display and EXIT program" << endl;
        int choice;
        do
        {
                cout << "Enter choice:";
                cin >> choice;
                switch (choice)
                {
                case 1: obj.payingCar();
                        break;
                case 2: obj.NonPaying();
                        break;
                case ESC: obj.Display();
                        break;
                default: cout << "Invalid." << endl;
                }
        } while (choice != ESC);
        system("pause");
        return 0;
}
```

OUTPUT:



C:\Users\Zainab Israr\source\repos\LT3\Debug\LT3.exe

```
Press 1 for Paying Car
Press 2 for Non-Paying Car
Press ESC key for display and EXIT program
Enter choice:1
Enter choice:2
Enter choice:3
Invalid.
Enter choice:1
Enter choice:2
Enter choice:5
Invalid.
Enter choice:27
Total number of cars:4
Total money collected:1
Press any key to continue . . .
```

QNO:4

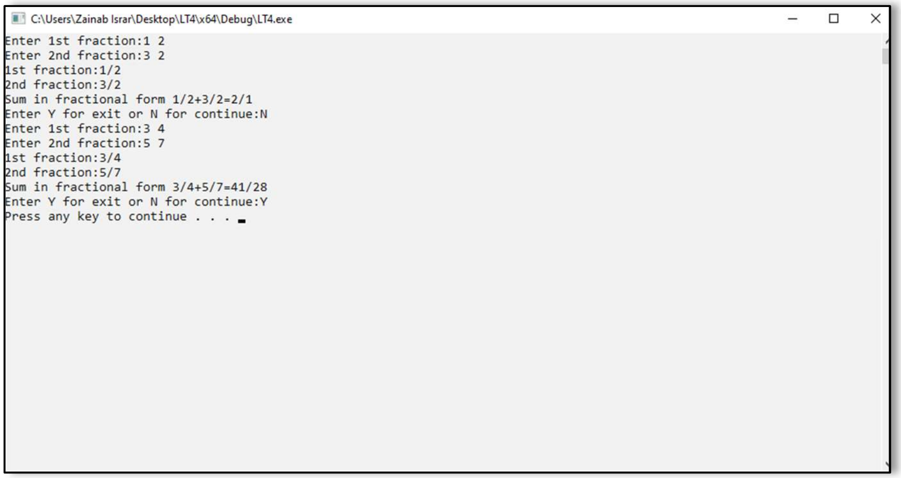
“Fraction.h”

```
#pragma once
#include<iostream>
using namespace std;
class Fraction
{
        int N1, D1, N2, D2; // N=Numerator , D=Denominator
public:
        void getFraction()
        {
            cout << "Enter 1st fraction:";
            cin >> N1 >> D1;
            cout << "Enter 2nd fraction:";
            cin >> N2 >> D2;
        }
        void DisplayFraction()
        {
            cout << "1st fraction:"<< N1<<"/"<< D1<<endl
               << "2nd fraction:"<< N2 <<"/"<<D2<<endl;
        }
        void SumInFractionalForm()
        {
            int n,d;
            n = ((N1*D2) + (N2*D1));
            d=(D1*D2);
            if (n%d == 0)
            {
                n /=d;
                d =1;
            }
      cout<<"Sum in fractional form "<<N1<<"/"<<D1<<"+"<< N2 << "/" << D2
<<"="<<n<<"/"<<d<<endl;
        }
};
```

“Fraction.cpp”

```
#include "Fraction.h"
#include<iostream>
using namespace std;
int main()
{
    Fraction obj;
    char c;
    do
    {
        obj.getFraction();
        obj.DisplayFraction();
        obj.SumInFractionalForm();
        cout << "Enter Y for exit or N for continue:";
        cin >>c;
    } while (c!='Y');
    system("pause");
    return 0;
}
```

OUTPUT:



C:\Users\Zainab Israr\Desktop\LT4\x64\Debug\LT4.exe

```
Enter 1st fraction:1 2
Enter 2nd fraction:3 2
1st fraction:1/2
2nd fraction:3/2
Sum in fractional form 1/2+3/2=2/1
Enter Y for exit or N for continue:N
Enter 1st fraction:3 4
Enter 2nd fraction:5 7
1st fraction:3/4
2nd fraction:5/7
Sum in fractional form 3/4+5/7=41/28
Enter Y for exit or N for continue:Y
Press any key to continue . . .
```

QNO:5

“Rectangle.h”

```
#pragma once
#include<iostream>
using namespace std;
class Rectangle
{
        float length, width;
public:
        Rectangle()
        {
                length = 1.0;
                width = 1.0;
        }
        float CalculatePerimeter()
        {
                float Perimeter;
                Perimeter =2*(length + width);
                return Perimeter;;
        }
        float CalculateArea()
        {
                int Area;
                Area = length * width;
                return Area;
        }
```

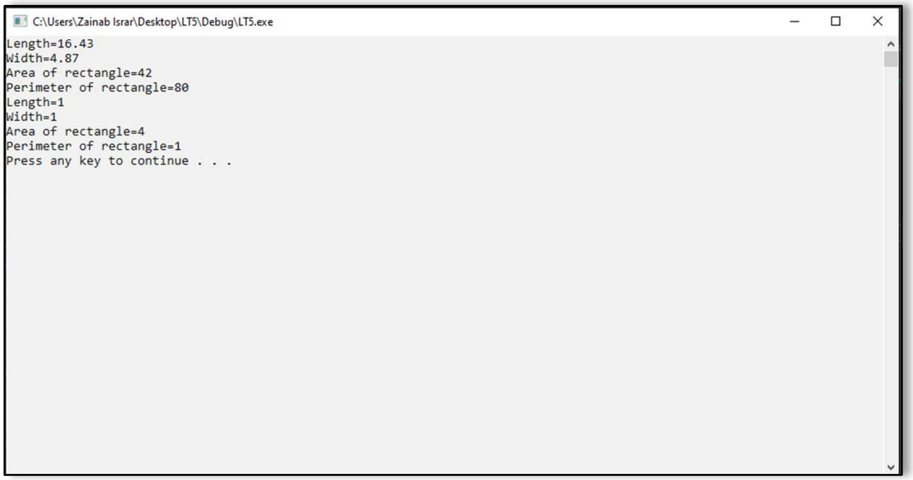
```
    float getLength()
    {
        return length;
    }
    float getWidth()
    {
        return width;
    }
    void setLength(float l)
    {
        if ((l> 0.0) && (l < 20.0))
            length = l;
    }
    void setWidth(float w)
    {
        if ((w> 0.0) && (w < 20.0))
            width = w;
    }
    void display(float a , float p)
    {
        cout << "Length=" << length << endl
            << "Width=" << width << endl
            << "Area of rectangle=" << a<< endl
            << "Perimeter of rectangle=" <<p<< endl;
    }
};
```

“Rectangle.cpp”

```
#include "Rectangle.h"
#include<iostream>
using namespace std;
int main()
{
    Rectangle obj1,obj2;
    obj1.getLength();
    obj1.getWidth();
    obj1.setLength(16.43);
    obj1.setWidth(4.87);
    int a=obj1.CalculatePerimeter();
    int b=obj1.CalculateArea();
    obj1.display(a,b);
    //check values that are out of range
    obj2.getLength();
    obj2.getWidth();
    obj2.setLength(21.43);
    obj2.setWidth(65.87);
```

```
	int x=obj2.CalculatePerimeter();
	int y=obj2.CalculateArea();
	obj2.display(x,y);
	system("pause");
	return 0;
}
```

OUTPUT:



```
Length=16.43
Width=4.87
Area of rectangle=42
Perimeter of rectangle=80
Length=1
Width=1
Area of rectangle=4
Perimeter of rectangle=1
Press any key to continue . . .
```

QNO:6

<u>“Time.h”</u>

```
#pragma once
#include<iostream>
using namespace std;
class Time
{
	int hours, minutes, second;
public:
	Time()
	{
		hours = 0;
		minutes = 0;
		second = 0;
	}
	Time(int h, int m, int s)
	{
		hours = h;
```

```
    minutes = m;
    second = s;
}
int  HoursIsValid(int h)
{
    if (h >= 0 && h <= 24)
    {
        hours = h;
        return h;
    }
}
int  MinutesIsValid(int m)
{
    if (m >= 0 && m <= 60)
    {
        minutes = m;
        return m;
    }
}
int SecondIsValid(int s)
{
    if (s >= 0 && s <= 60)
    {
        second = s;
        return s;
    }
}
void setTime(int h, int m, int s)
{
    hours = h;
    minutes = m;
    second = s;
}
void increment()
{
    second++;
    if (second ==60)
    {
        minutes++;
        second = 0;
    }
    if (minutes == 60)
    {
        hours++;
        minutes = 0;
    }
```

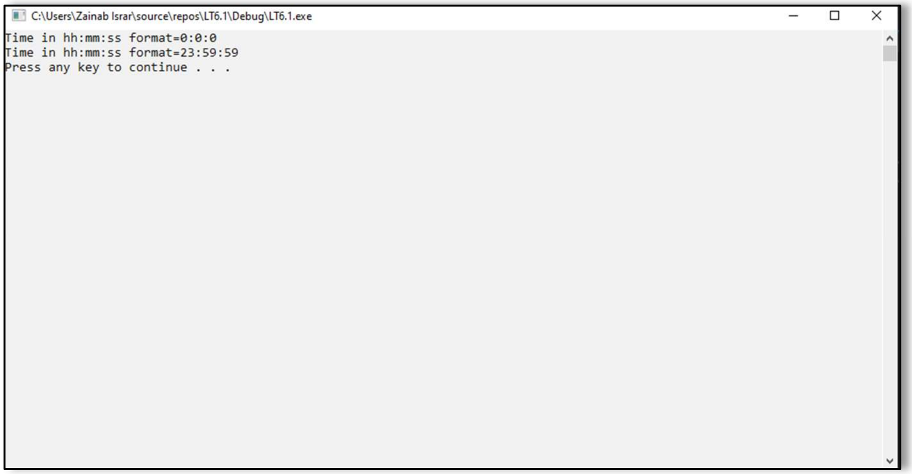
```
        if (hours == 24)
        {
          hours = 0;
        }
    }
    void decrement()
    {
        second--;
        if (second <= 0)
        {
            minutes--;
            second = 59;
        }
        if (minutes <= 0)
        {
            hours--;
            minutes = 59;
        }
        if (hours <= 0)
        {
            hours = 23;
        }
    }
    void display()
    {
        cout << "Time in hh:mm:ss format=" << hours << ":" << minutes << ":" <<
second << endl;
    }
};
```

“Time.cpp”

```
#include "Time.h"
#include<iostream>
using namespace std;
int main()
{
    Time obj1;
    Time obj2(0, 0, 0);
    int a=obj1.HoursIsValid(23);
    int b = obj1.MinutesIsValid(59);
    int c = obj1.SecondIsValid(59);
    obj1.setTime(a, b, c);
    // no change occur if both increment and decrement apply on 1 time
    obj1.increment();
    obj1.display();
    obj2.decrement();
```

```
	obj2.display();
	system("pause");
	return 0;
}
```

OUTPUT:



```
Time in hh:mm:ss format=0:0:0
Time in hh:mm:ss format=23:59:59
Press any key to continue . . .
```

QNO:7

“Date.h”

```
#pragma once
#include<iostream>
using namespace std;
class Date
{
	int day, month, year;
public:
	Date()
	{
		day = 0;
		month = 0;
		year = 0;
	}
	Date(int d, int m, int y)
	{
		day = d;
		month = m;
		year = y;
}
int DayIsValid(int d)
{
        if (d >= 0 && d <= 31)
        {
                day = d;
                return d;
        }
}
int MonthIsValid(int m)
{
        if (m >= 0 && m <= 12)
        {
                month = m;
                return m;
        }
}
int YearIsValid(int y)
{
        if (y >= 0000)
        {
                year = y;
                return y;
        }
}
void setDate(int d, int m,int y)
{
        day = d;
        month = m;
        year = y;
}
void increment()
{
        day++;
        if (day == 31 || day == 30 || day == 29 || day == 28)
        {
                month++;
                day = 1;
        }
        if (month == 12)
        {
                year++;
                month = 1;
        }
}
void decrement()
```

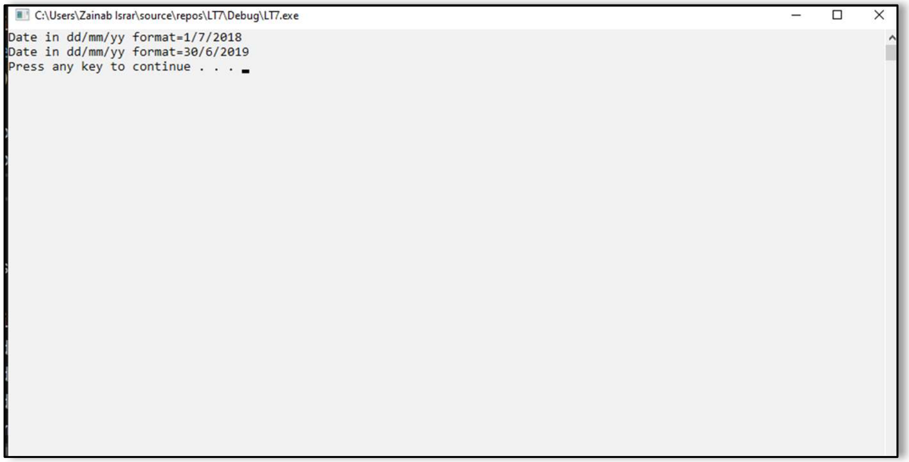
```
    {
        day--;
        if (day <=1)
        {
            month--;
            if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8 ||
month == 10 || month == 12)
                day = 31;
            else if (month == 4 || month == 6 || month == 9 || month == 11)
                day = 30;
            else if (month==2 && year % 4 == 0 || year % 100 == 0 || year % 400 ==
0) //to check leap year
                day = 29;
            else
                day = 28;
        }
        if (month <= 1)
        {
            year--;
            month =12;
        }
    }
    void dispaly()
    {
        cout << "Date in dd/mm/yy format=" << day << "/" << month << "/" << year <<
endl;
    }
};
```

“Date.cpp”

```
#include "Date.h"
#include<iostream>
using namespace std;
int main()
{
    Date obj1;
    Date obj2(01, 07, 2019);
    int a = obj1.DayIsValid(30);
    int b = obj1.MonthIsValid(06);
    int c = obj1.YearIsValid(2018);
    obj1.setDate(a, b, c);
    // no change occur if both increment and decrement apply on 1 date
    obj1.increment();
    obj1.dispaly();
    obj2.decrement();
    obj2.dispaly();
```

```
	system("pause");
	return 0;
}
```

OUTPUT:



```
Date in dd/mm/yy format=1/7/2018
Date in dd/mm/yy format=30/6/2019
Press any key to continue . . .
```

QNO:8

<u>“Date.h”</u>

```
#pragma once
 #include<iostream>
using namespace std;
class Date
{
public:
	int year1, month1, date1 ,year2 ,month2 ,date2;
	Date()
	{
		year1= 0;
		month1= 0;
		date1= 0;
		year2= 0;
		month2= 0;
		date2= 0;
	}
	void Enter()
	{
		cout << "Enter 1st  date :";
		cin >> date1 >> month1 >> year1;
```

```
        cout << "Enter 2nd  date :";
        cin >> date2 >> month2 >> year2;
    }
    void addition()
    {
        int year, month, date;
        if ((year1 >= 0001 && year1 <= 9999) && (month1 >= 1 && month1 <= 12)
&& (date1 >= 1 && date1 <= 30))
        {
            if ((year2 >= 0001 && year2 <= 9999) && (month2 >= 1 && month2 <=
12) && (date2 >= 1 && date2 <= 30))
            {
                if (year1 == year2)
                    year = year1; //year=year2;
                else if (year1 != year2)
                    year = year1 +year2;
                if (month1 == month2)
                    month = month1; //month=month2;
                else if (month1 != month2)
                    month = month1 +month2;
                  if (date1 == date2)
                    date = date1; // date = date2;
                else if (date1!= date2)
                    date = date1 +date2;
                if (date > 30)
                {
                    date-=30;
                    month++;
                }
                if (month > 12)
                {
                    month-=12;
                    year++;
                }
                cout << "Addition of two dates:" << date << "/" << month << "/"
<< year << endl;
            }
        }
    }
    void subtraction()
    {
        int year, month, date;
        if ((year1 >= 0001 && year1 <= 9999) && (month1 >= 1 && month1 <= 12)
&& (date1 >= 1 && date1 <= 30))
        {
            if ((year2 >= 0001 && year2 <= 9999) && (month2 >= 1 && month2 <=
12) && (date2 >= 1 && date2 <= 30))
            {
                if (year1 == year2)
                    year = year1; //year=year2;
                else if (year1 > year2)
                    year = year1 - year2;
                else if (year1 < year2)
                    year = year2 - year1;
                if (month1 == month2)
                    month = month1; //month=month2;
                else if (month1 > month2)
                    month = month1 - month2;
                else if (month1 < month2)
                    month = month2 - month1;
                if (date1 == date2)
                    date = date1; // date = date2;
                else if (date1 > date2)
                    date = date1 - date2;
                else if (date1 < date2)
                    date = date2 - date1;
            }
        }
        cout << "Subtraction of two dates:" << date << "/" << month << "/" << year <<
endl;
    }
};
```

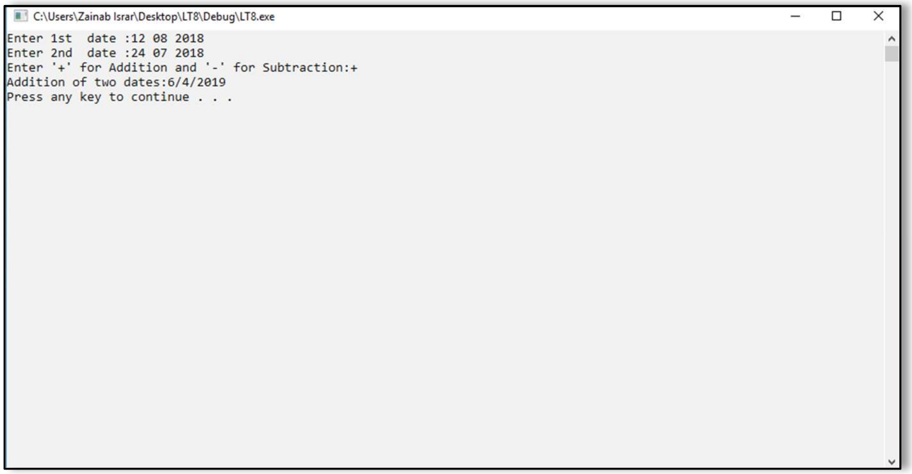
“Date.cpp”

```
#include "Date.h"
#include<iostream>
using namespace std;
int main()
{
    Date obj;
    obj.Enter();
    char c;
    cout << "Enter '+' for Addition and '-' for Subtraction:";
    cin >> c;
    switch (c)
    {
     case '+': obj.addition();
    break;
     case '-': obj.subtraction();
         break;
     default: cout << "Invalid." << endl;
    }
```

```
        system("pause");
        return 0;
}
```

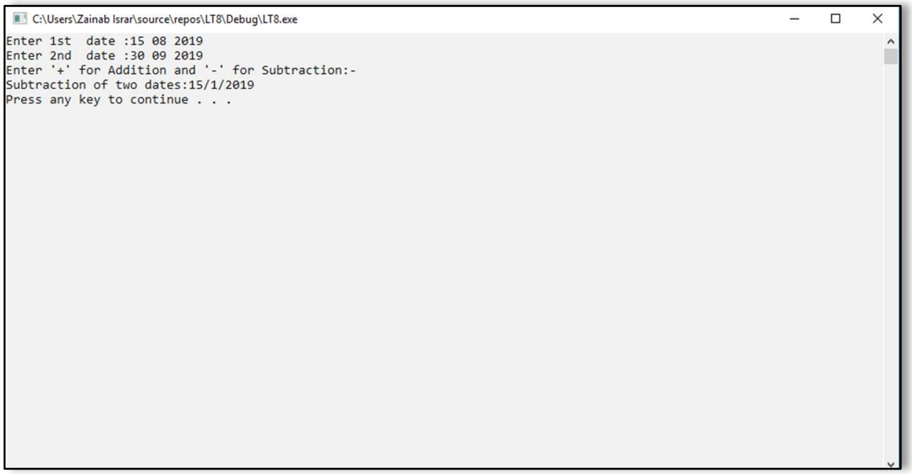
OUTPUT:

Addition:



```
Enter 1st  date :12 08 2018
Enter 2nd  date :24 07 2018
Enter '+' for Addition and '-' for Subtraction:+
Addition of two dates:6/4/2019
Press any key to continue . . .
```

Subtraction:



```
Enter 1st  date :15 08 2019
Enter 2nd  date :30 09 2019
Enter '+' for Addition and '-' for Subtraction:-
Subtraction of two dates:15/1/2019
Press any key to continue . . .
```

QNO:9

“Student.h”

```
#pragma once
#include<iostream>
#include<string>
using namespace std;
class Student
{
	int ID, CreditHOurs[5];
	string name,batch;
	double obtainedGradePoints[5];
public:
	Student();
	void getInfo();
	double GPA();
	void displayInfo(double gpa);
};
```

“Student.cpp”

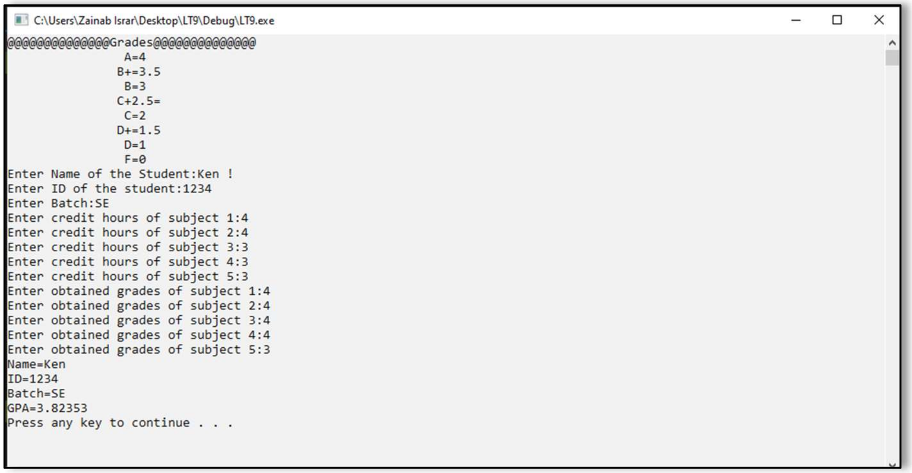
```
#include "Student.h"
#include<iostream>
#include<string>
using namespace std;
Student::Student()
{
	ID = 0;
	name =" ";
	batch =" ";
	for (int i = 0; i < 5; i++)
	{
		CreditHOurs[i] = 0;
		obtainedGradePoints[i] = 0.0;
	}
}
void Student::getInfo()
{
	cout << "Enter Name of the Student:";
	getline(cin, name, '!');
	cout << "Enter ID of the student:";
	cin >> ID;
	cout << "Enter Batch:";
	cin >> batch;
	for (int i = 0; i < 5; i++)
	{
```

```
                cout << "Enter credit hours of subject " << i + 1<<":";
                cin >> CreditHOurs[i];
        }
        for (int i = 0; i < 5; i++)
        {
                cout << "Enter obtained grades of subject " << i + 1<<":";
                cin >> obtainedGradePoints[i];
        }
}
double Student::GPA()
{
        double x=0.0,gpa;
        int y=0;
        for (int i = 0; i < 5; i++)
                y += CreditHOurs[i];
        for (int i = 0; i < 5; i++)
                 x+= CreditHOurs[i] * obtainedGradePoints[i];
        gpa = x / y;
        return gpa;
}
void Student::displayInfo(double gpa)
{
        cout << "Name=" << name << endl
                << "ID=" << ID << endl
                << "Batch=" << batch << endl
                << "GPA=" << gpa << endl;
}
```

“Source.cpp”

```
#include"Student.h"
#include<iostream>
using namespace std;
int main()
{
        cout << "@@@@@@@@@@@@@@Grades@@@@@@@@@@@@@@" <<
endl
                << "                A=4" << endl
                << "               B+=3.5" << endl
                << "                B=3" << endl
                << "               C+2.5=" << endl
                << "                C=2" << endl
                << "               D+=1.5" << endl
                << "                D=1" << endl
                << "                F=0" << endl;
        Student obj;
        obj.getInfo();
        double gpa;
        gpa = obj.GPA();
        obj.displayInfo(gpa);
        system("pause");
        return 0;
}
```

OUTPUT:



```
@@@@@@@@@@@@@@Grades@@@@@@@@@@@@@@
                A=4
               B+=3.5
                B=3
               C+2.5=
                C=2
               D+=1.5
                D=1
                F=0
Enter Name of the Student:Ken !
Enter ID of the student:1234
Enter Batch:SE
Enter credit hours of subject 1:4
Enter credit hours of subject 2:4
Enter credit hours of subject 3:3
Enter credit hours of subject 4:3
Enter credit hours of subject 5:3
Enter obtained grades of subject 1:4
Enter obtained grades of subject 2:4
Enter obtained grades of subject 3:4
Enter obtained grades of subject 4:4
Enter obtained grades of subject 5:3
Name=Ken
ID=1234
Batch=SE
GPA=3.82353
Press any key to continue . . .
```

QNO:10

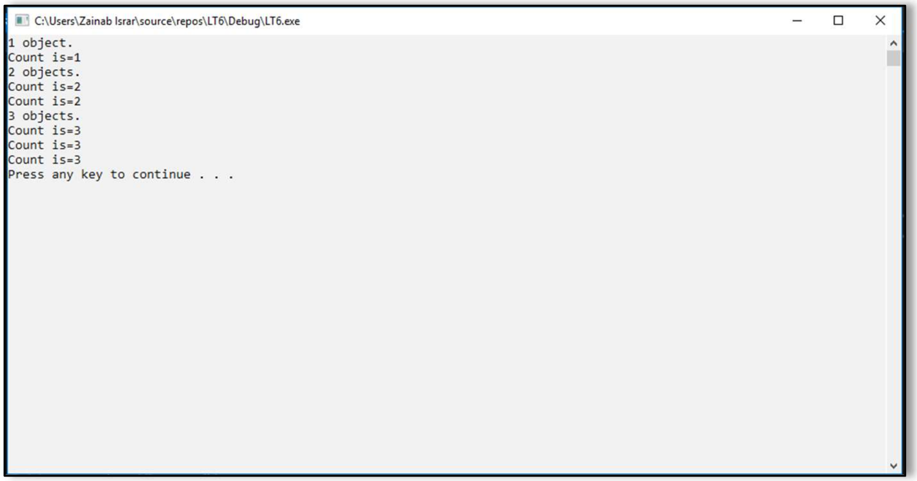
<u>“Record.h”</u>

```
#pragma once
#include<iostream>
using namespace std;
class Record
{
        static int count;
public:
        Record()
        {
                count++;
        }
        int getCount()
        {
                return count;
        }
};
//int Record::count = 0;
```

“Record.cpp”

```
#include "Record.h"
#include<iostream>
using namespace std;
int Record::count = 0;
int main()
{
	Record obj1;
	cout << "1 object." << endl;
	cout << "Count is=" << obj1.getCount()<<endl;
	Record obj2;
	cout << "2 objects." << endl;
	cout << "Count is=" << obj1.getCount() << endl;
	cout << "Count is=" << obj2.getCount() << endl;
	Record obj3;
	cout << "3 objects." << endl;
	cout << "Count is=" << obj1.getCount() << endl;
	cout << "Count is=" << obj2.getCount() << endl;
	cout << "Count is=" << obj3.getCount() << endl;
	system("pause");
	return 0;
}
```

OUTPUT:



```
1 object.
Count is=1
2 objects.
Count is=2
Count is=2
3 objects.
Count is=3
Count is=3
Count is=3
Press any key to continue . . .
```

QNO:11

“ComplexValue.h”

```
#pragma once
#include<iostream>
```

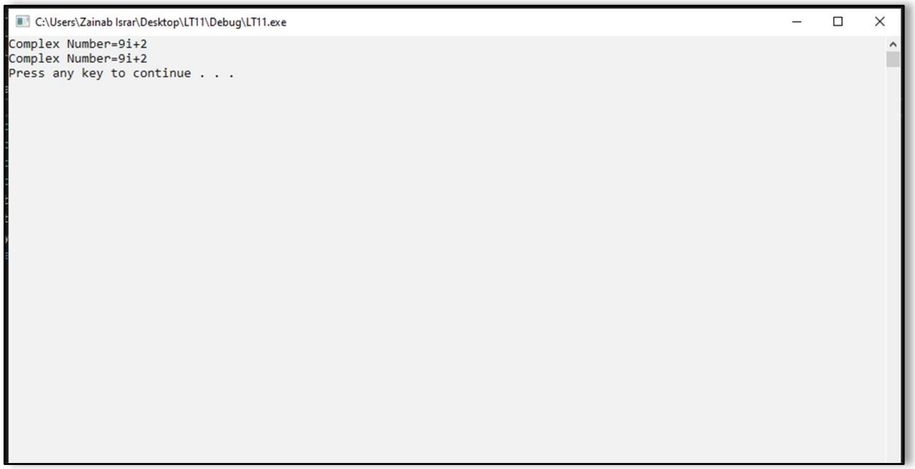
```
#include<string>
using namespace std;
class ComplexValue
{
        int real;
        string imaginary;
public:
        ComplexValue()
        {
                real = 0;
                imaginary =" ";
        }
        ComplexValue(int r)
        {
                real = r;
        }
        ComplexValue(string i)
        {
                imaginary = i;
        }
        ComplexValue(int r, string i)
        {
                real = r;
                imaginary = i;
        }
        void display()
        {
                cout << "Complex Number=" << imaginary << "+" << real << endl;
        }
};
```

“ComplexValue.cpp”

```
#include "ComplexValue.h"
#include<iostream>
#include<string>
using namespace std;
int main()
{
        ComplexValue obj1(5);
        ComplexValue obj2("3i");
        ComplexValue obj3(2,"9i");
        ComplexValue obj4(obj3);  //Default Copy Constructor
        obj3.display();
        obj4.display();
```

```
	system("pause");
	return 0;
}
```

OUTPUT:



```
Complex Number=9i+2
Complex Number=9i+2
Press any key to continue . . .
```

QNO:12

“Student.h”

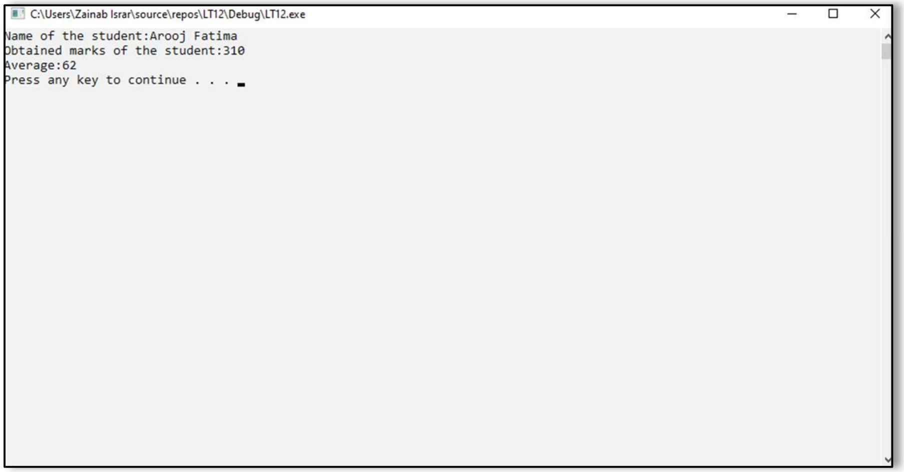
```
#pragma once
#include<iostream>
#include<string>
using namespace std;
class Student
{
	string sname;
	int marks[5], obtMarks, totalMarks;
public:
	void assign()
	{
		sname = "Arooj Fatima";
		marks[0] = 45;
		marks[1] = 76;
		marks[2] = 89;
		marks[3] = 34;
		marks[4] = 66;
		obtMarks = 0;
		totalMarks = 500;
```

```
            }
    double compute()
    {
            for (int i = 0; i < 5; i++)
                    obtMarks += marks[i];
                    double average = obtMarks/5;
                    return average;
    }
    void display(double average)
    {
            cout << "Name of the student:" << sname << endl
                    << "Obtained marks of the student:" << obtMarks<<endl
                    << "Average:" << average << endl;
    }
};
```

<u>“Student.cpp”</u>

```
#include "Student.h"
#include<iostream>
#include<string>
using namespace std;
int main()
{
        Student obj;
        obj.assign();
        double avg=obj.compute();
        obj.display(avg);
        system("pause");
        return 0;
}
```

OUTPUT:



```
Name of the student:Arooj Fatima
Obtained marks of the student:310
Average:62
Press any key to continue . . .
```

QNO:13

“Bakery.h”

```
#pragma once
#include<iostream>
#include<string>
using namespace std;
class Bakery
{
	int id,cost, quantity;
	string name;
	static int productID;
public:
	Bakery()
	{
		cost = 0.0;
		name = "";
		quantity = 0.0;
		productID++;
		id = productID;
	}
	void displayInfo()
	{
		cout << "Name of the product:" << name << endl
			<< "Quantity of the product:" << quantity << endl
			<< "Cost of the product:" << cost << endl
			<< "ProductID:" << id << endl;
	}
	void BuyProduct(Bakery obj[5])
	{
		string n;
		cout << "Enter name of product which you want to buy:";
		cin >> n;
		for (int i = 0; i < 5; i++)
		{
			if (obj[i].name == n)
				cout << "Product is Available." << endl;
		}
	}
	void AddItem()
	{
		cout << "Enter name of product:";
		cin>>name;
		cout << "Enter quantity of product:";
		cin >> quantity;
		cout << "Enter cost of product:";
```

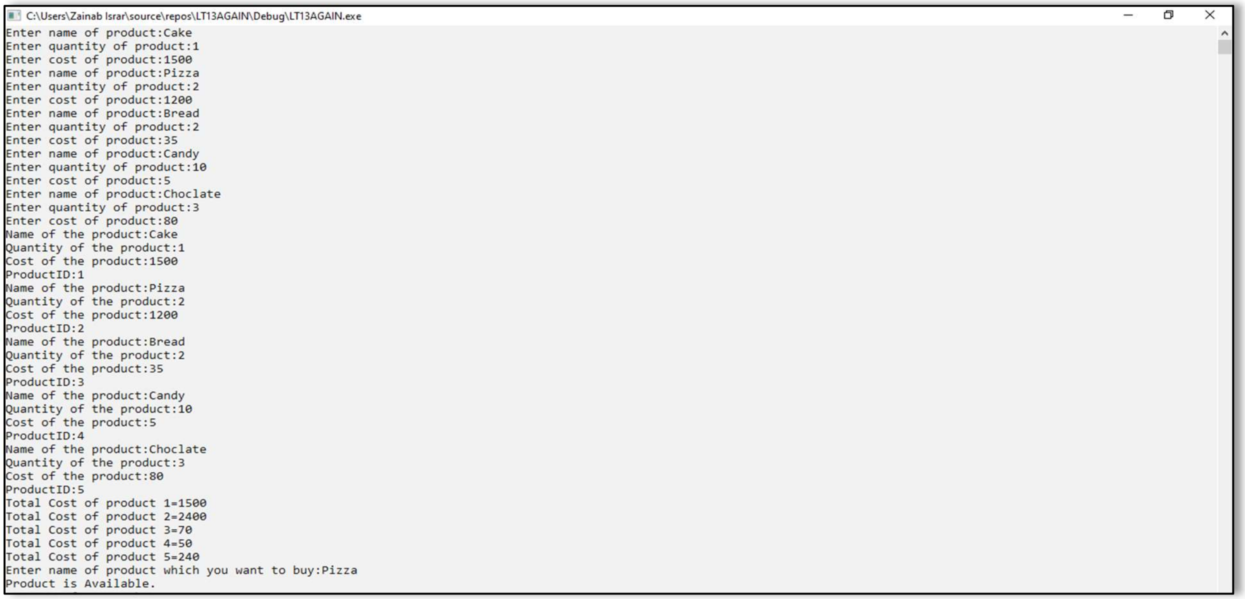
```
                cin >> cost;
        }
        void FindProductByID(Bakery obj[5])
        {
                int ID;
                cout << "Enter ID for search:";
                cin >> ID;
                for (int i = 0; i < 5; i++)
                {
                        if (obj[i].id == ID)
                        {
                                cout << "Name=" << obj[i].name << endl;
                                cout << "ID=" << obj[i].id << endl;
                                cout << "Cost=" << obj[i].cost << endl;
                                cout << "Quantity=" << obj[i].quantity << endl;
                        }
                }
        }
        void TotalCost(Bakery obj[5])
        {
                double totalCost;
                for (int i = 0; i < 5; i++)
                {
                        totalCost = obj[i].cost * obj[i].quantity;
                        cout << "Total Cost of product " << i + 1 <<"="<<totalCost << endl;
                }
        }
};
```

“Bakery.cpp”

```
#include "Bakery.h"
#include<iostream>
#include<string>
using namespace std;
int Bakery::productID = 0;
int main()
{
        Bakery B[5],obj;
        for (int i = 0; i < 5; i++)
        {
                B[i].AddItem();
        }
        for (int i = 0; i < 5; i++)
        {
                B[i].displayInfo();
        }
```

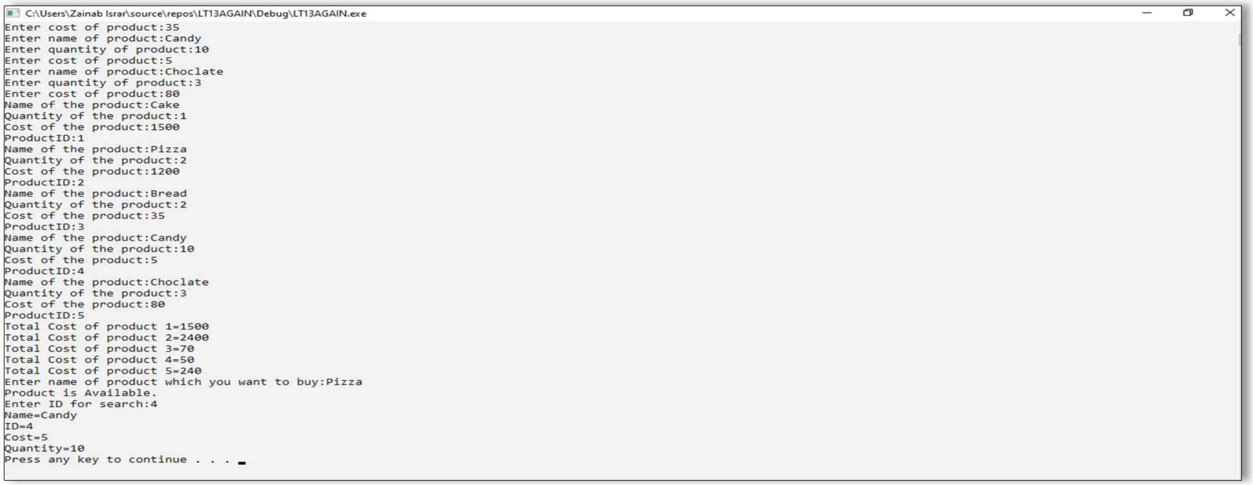
```
         obj.TotalCost(B);
         obj.BuyProduct(B);
         obj.FindProductByID(B);
        system("pause");
        return 0;
}
```

OUTPUT:



C:\Users\Zainab Israr\source\repos\LT13AGAIN\Debug\LT13AGAIN.exe

```
Enter name of product:Cake
Enter quantity of product:1
Enter cost of product:1500
Enter name of product:Pizza
Enter quantity of product:2
Enter cost of product:1200
Enter name of product:Bread
Enter quantity of product:2
Enter cost of product:35
Enter name of product:Candy
Enter quantity of product:10
Enter cost of product:5
Enter name of product:Choclate
Enter quantity of product:3
Enter cost of product:80
Name of the product:Cake
Quantity of the product:1
Cost of the product:1500
ProductID:1
Name of the product:Pizza
Quantity of the product:2
Cost of the product:1200
ProductID:2
Name of the product:Bread
Quantity of the product:2
Cost of the product:35
ProductID:3
Name of the product:Candy
Quantity of the product:10
Cost of the product:5
ProductID:4
Name of the product:Choclate
Quantity of the product:3
Cost of the product:80
ProductID:5
Total Cost of product 1=1500
Total Cost of product 2=2400
Total Cost of product 3=70
Total Cost of product 4=50
Total Cost of product 5=240
Enter name of product which you want to buy:Pizza
Product is Available.
```



C:\Users\Zainab Israr\source\repos\LT13AGAIN\Debug\LT13AGAIN.exe

```
Enter cost of product:35
Enter name of product:Candy
Enter quantity of product:10
Enter cost of product:5
Enter name of product:Choclate
Enter quantity of product:3
Enter cost of product:80
Name of the product:Cake
Quantity of the product:1
Cost of the product:1500
ProductID:1
Name of the product:Pizza
Quantity of the product:2
Cost of the product:1200
ProductID:2
Name of the product:Bread
Quantity of the product:2
Cost of the product:35
ProductID:3
Name of the product:Candy
Quantity of the product:10
Cost of the product:5
ProductID:4
Name of the product:Choclate
Quantity of the product:3
Cost of the product:80
ProductID:5
Total Cost of product 1=1500
Total Cost of product 2=2400
Total Cost of product 3=70
Total Cost of product 4=50
Total Cost of product 5=240
Enter name of product which you want to buy:Pizza
Product is Available.
Enter ID for search:4
Name=Candy
ID=4
Cost=5
Quantity=10
Press any key to continue . . .
```