PPL Assignment 1

Name: Himani Verma

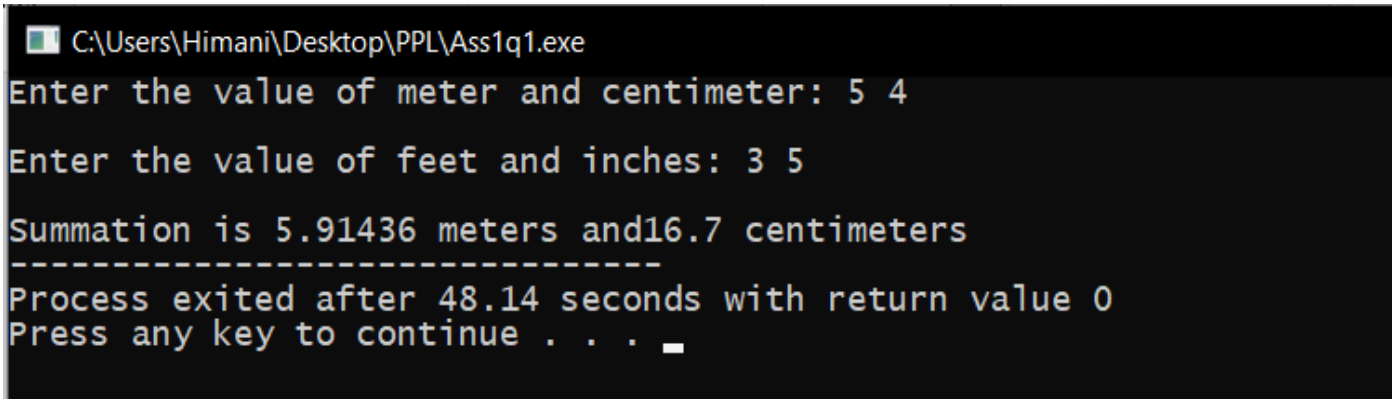
Admission No: U19CS075

1. Create two classes DM and DB which store the value of distances. DM stores distances in metres and centimetres and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a function to carry out the addition operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or metres and centimetres depending on the object on display.

Source Code:

```
#include<iostream>
using namespace std;
#include<conio.h>
#include<stdio.h>
#include<math.h>

class DM{
    public:
        double meter, centimeter;
};

class DB{
    public:
        double feet,inches;
};

void add(DM dm, DB db){
    double d1,d2;
    cout<<"Enter the value of meter and centimeter: ";
    cin>>dm.meter>>dm.centimeter;
    cout<<"\nEnter the value of feet and inches: ";
    cin>>db.feet>>db.inches;
    dm.meter=dm.meter+(db.feet)*0.305;
    cout<<"\nSummation is "<<dm.meter <<" meters and ";
    dm.centimeter=dm.centimeter+(db.inches)*2.54;
    cout<<dm.centimeter<<" centimeters";
}
int main(){
    DM dm; //Creating Objects
    DB db;
    add(dm,db);
}
```

Output:



2. Find errors, if any, in the following C++ statements.
    1. long float x;
    Error: Too many types
    Correction: float x; or double x;

    2. char *cp = vp; // vp is a void pointer
    Error: Type must be matched
    Correction:char *cp = (char*) vp;

    3. int code = three; // three is an enumerator
    Error: no error

    4. int sp = new; // allocate memory with new
    Error: syntax error
    Correction: int*p=new int[10];

    5. enum (green, yellow, red);
    Error: tag name missing
    Correction: enum color (green, yellow, red)

    6. int const sp = total;
    Error: address have to assign instead of content
    Correction: int const*p = &total;

    7. const int array_size;
    Error: C++ requires a const to be initialized
    Correction: const int array-size = 5;

    8. for (i=1; int i<10; i++) cout << i << "/n";
    Error: undefined symbol i
    Correction: for(int i=1;i<10;i++)cout<<i<<"/n";

    9. int & number = 100;
    Error: invalid variable name
    Correction: int number = 100;

    10. float *p = new int 1101;
    Error: wrong data type
    Correction: float *p = new float[10];

11. int public = 1000;
Error: keyword can not be used as a variable name
Correction: int public1 = 1000;

12. char name[33] = "USA";
Error: array size of char must be larger than the number of characters in the string
Correction: chat name[4] = "USA";

3. Assume that a bank maintains two kinds of accounts for customers, one called a savings account and the other as a current account. The savings account provides simple interest and withdrawal facilities but no cheque book facility. The current account provides a check book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.
Create a class account that stores customer name, account number and type of account. From this derive the classes cur_acct and sav_acct to make them more specific to their requirements. Include necessary member functions in order to achieve the following tasks:

a. Accept deposits from a customer and update the balance.
b. Display the balance.
c. Compute and deposit interest.
d. Permit withdrawal and update the balance.
e. Check for the minimum balance, impose penalty, necessary and update the balance.
f. Do not use any constructors. Use member functions to initialize the class members.

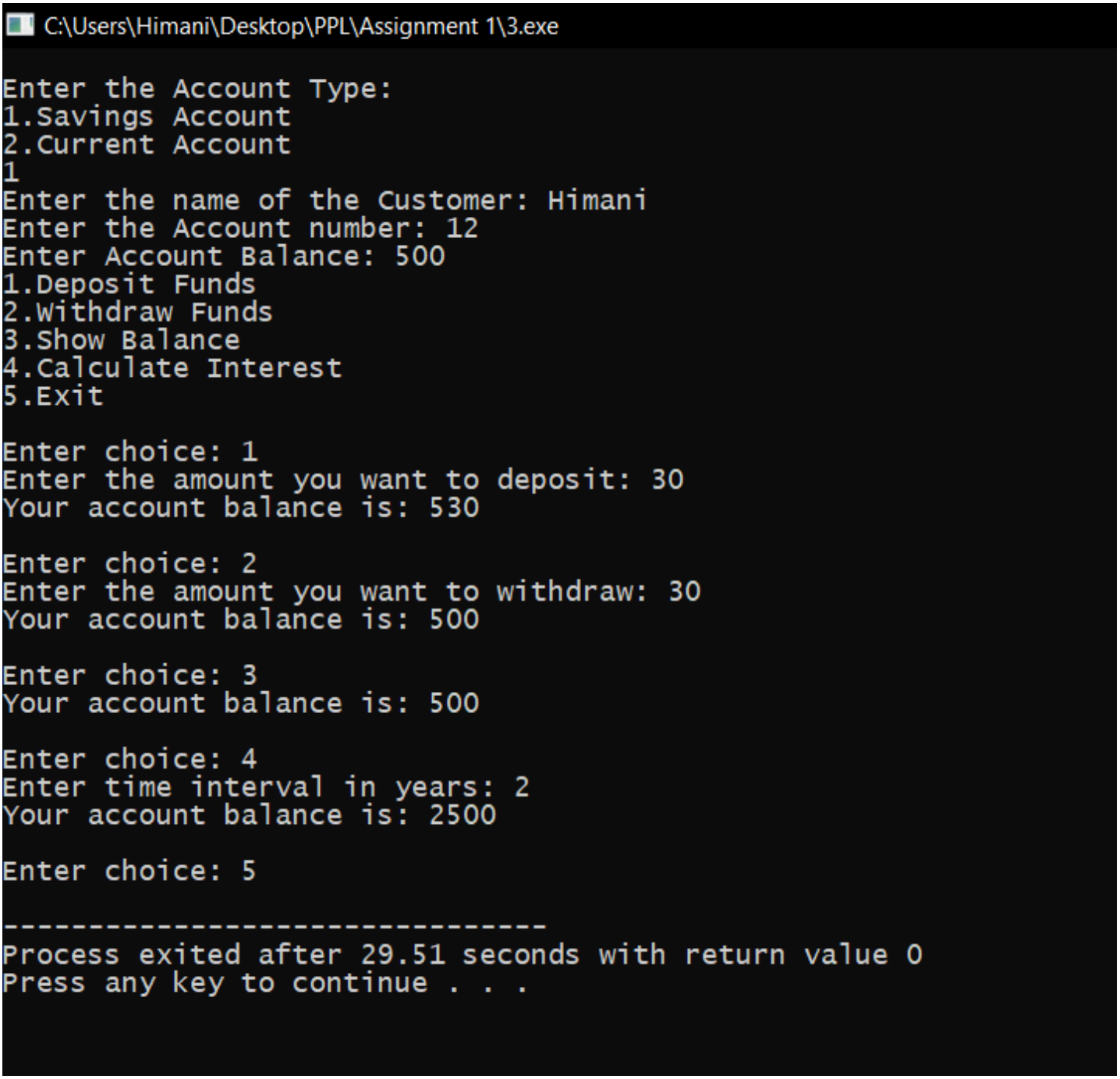
Source Code:

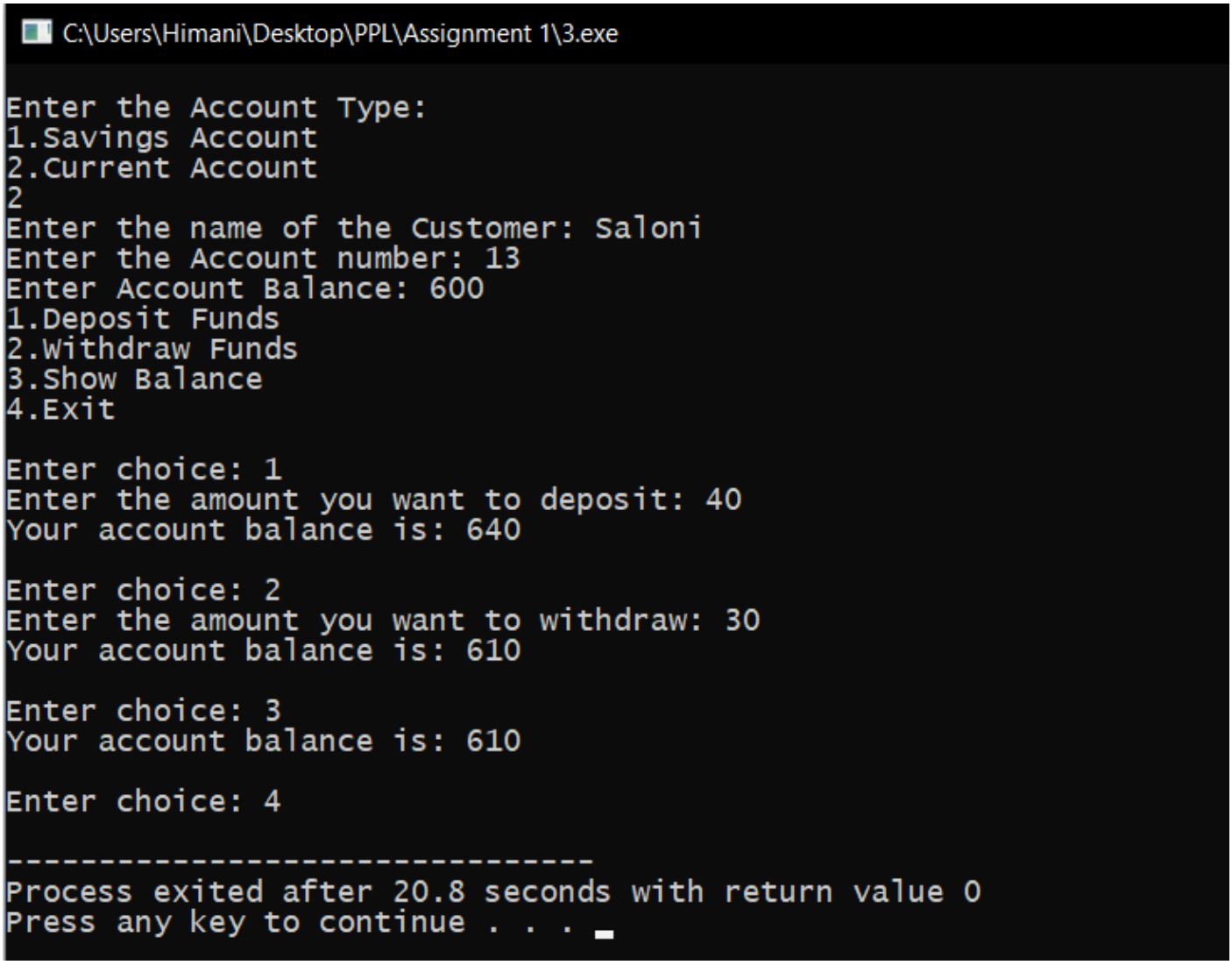
```
#include<iostream>
#include<stdlib.h>
using namespace std;

class account {
    public:
        int acc_num,acc_type,balance=0,amount=0;
        char customer_name[20];

        void getData(int type){
            cout<<"Enter the name of the Customer: ";
            cin>>customer_name;
            cout<<"Enter the Account number: ";
            cin>>acc_num;
            cout<<"Enter Account Balance: ";
            cin>>balance;
            acc_type=type;
        }

        void display(){
            cout<<"Your account balance is: "<<balance<<endl;
        }

        void deposit(){
            cout<<"Enter the amount you want to deposit: ";
```

```
            cin>>amount;
            balance=balance+amount;
            display();
        }

        void withdraw(){
            cout<<"Enter the amount you want to withdraw: ";
            cin>>amount;
            if(amount>balance){
                cout<<"Insufficient Funds\n";
            }
            else{
                balance=balance-amount;
            }
            display();
        }

};

class current: public account{
    public:
        void penalty(){
            if(balance<500 && acc_type==2){
                cout<<"Balance is less than 500 so penalty will be imposed.";
                balance=balance-20;
                if(balance<0) balance=0;
                display();
            }
        }
};

class savings: public account{
    public:
        void interest(){
            int t;
            cout<<"Enter time interval in years: ";
            cin>>t;
            balance=balance*(1+2*t);
            display();
        }
};


int main(){
    int type;
    cout<<"\nEnter the Account Type: \n1.Savings Account \n2.Current Account\n";
    cin>>type;
    int ch;
    if (type==1){
        savings ac;
        ac.getData(type);
        cout<<"1.Deposit Funds\n"
        <<"2.Withdraw Funds\n"
        <<"3.Show Balance\n"
```

```
        <<"4.Calculate Interest\n"
        <<"5.Exit\n";

        while(true){
        cout<<"\nEnter choice: ";
        cin>>ch;
        switch (ch)
        {
            case 1:
                ac.deposit(); break;
            case 2:
                ac.withdraw(); break;
            case 3:
                ac.display();
                break;
            case 4:
                ac.interest();
                break;
            case 5:
                exit(0);
            default:
                break;
            }
        }
    }else{
        current ac;
        ac.getData(type);
        cout<<"1.Deposit Funds\n"<<"2.Withdraw Funds\n"
        <<"3.Show Balance\n"<<"4.Exit\n";

        while(true){
            cout<<"\nEnter choice: ";
            cin>>ch;
            switch (ch)
            {
            case 1:
                ac.deposit(); break;
            case 2:
                ac.withdraw();
                ac.penalty();
                break;
            case 3:
                ac.display();
                break;
            case 4:
                exit(0);
            default:
                break;
            }
        }
    }

    return 0;
}
```

Output:


C:\Users\Himani\Desktop\PPL\Assignment 1\3.exe
Enter the Account Type:
1.Savings Account
2.Current Account
1
Enter the name of the Customer: Himani
Enter the Account number: 12
Enter Account Balance: 500
1.Deposit Funds
2.Withdraw Funds
3.Show Balance
4.Calculate Interest
5.Exit
Enter choice: 1
Enter the amount you want to deposit: 30
Your account balance is: 530
Enter choice: 2
Enter the amount you want to withdraw: 30
Your account balance is: 500
Enter choice: 3
Your account balance is: 500
Enter choice: 4
Enter time interval in years: 2
Your account balance is: 2500
Enter choice: 5
Process exited after 29.51 seconds with return value 0
Press any key to continue . . .



4. An educational institution wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in the following figure. The figure also shows the minimum information required for each class. Specify all classes and define functions to create the database and retrieve individual information as and when required. The database created does not include educational information of the staff. It has been decided to add this information to teachers and officers (and not for typists) which will help management in decision making with regard to training, promotions etc. Add another data class called education that holds two pieces of educational information namely highest qualification in general education and highest professional qualification. This class should be inherited by the class's teacher and officer.

Source Code:

```
#include<iostream>
#include<stdlib.h>
#include<math.h>
#include<iomanip>
#include<stdio.h>
#include<conio.h>
#include<string.h>
using namespace std;

class staff{
        protected:
```

```
        int code;
        char name[30];
        public:
        void staff_info(const char *n, int c){
            strcpy(name,n);
            code=c;
        }
};
class education: public staff{
    protected:
        char highest[50],professional[50];
        public:
            void set_quali(const char *h,const char *p){
                strcpy(highest,h);
                strcpy(professional,p);
            }
};

class teacher: public education{
    protected:
        char subject[50],publication[50];
        public:
            void set_teacher(const char *s, const char *p){
                strcpy(subject,s);
                strcpy(publication,p);
            }
            void show(){
                cout<<"Name
"<<setw(8)<<"Code"<<setw(15)<<"Subject"<<setw(22)<<"Publication"<<setw(60)<<"Highest
qualification in general education"<<setw(40)<<"Highest professional qualification"<<endl
                <<name<<setw(8)<<code<<setw(20)<<subject<<setw(30)<<publication<<setw(25)<
<highest<<setw(60)<<professional<<endl;
            }
};
class officer: public education{
    protected:
        char grade[100];
        public:
            void set_officer(const char *g){
                strcpy(grade,g);
            }
        void show()
                {
                      cout<<"Name
"<<setw(8)<<"Code"<<setw(15)<<"Catagory"<<setw(60)<<"Highest qualification in general
education"<<setw(40)<<"Highest professional qualification"<<endl<<name<<setw(4)
                      <<code<<setw(25)<<grade<<setw(25)<<highest<<setw(65)<<professional<<
endl<<endl;
                }
};

class typist: public staff{
    protected:
        float speed;
```

```
        public:
            void set_typist(float s){
                speed=s;
            }
};

class regular: public typist{
    protected:
        float wage;
        public:
            void set_regular(float w){
            wage=w;
            }
            void show()
          {
                cout<<"Name
"<<setw(16)<<"Code"<<setw(15)<<"Speed"<<setw(15)<<"Wage"<<endl<<name<<setw(16)<<code
                <<setw(16)<<speed<<setw(16)<<wage<<endl<<endl;
          }
};

class casual: public typist{
    protected:
        float wage;
        public:
            void set_casual(float w){
            wage=w;
            }
            void show()
          {
                cout<<"Name
"<<setw(16)<<"Code"<<setw(15)<<"Speed"<<setw(15)<<"Wage"<<endl<<name<<setw(16)<<code
                <<setw(16)<<speed<<setw(16)<<wage<<endl<<endl;
          }

};

int main(){
    teacher t;
    cout<<"\nAbout teacher: "<<endl;
    t.staff_info("Aman",12);
    t.set_quali("BA","Diploma in Lets Code");
    t.set_teacher("Computer", "Lets C through fun");
    t.show();
    cout<<endl;
    t.staff_info("Kush",20);
    t.set_quali("BE","Masters in Geography");
    t.set_teacher("Social Studies", "How the world has changed");
    t.show();

    officer o;
    o.staff_info("Abhishek",14);
    o.set_quali("B.tech","Masters in Structural Engineering");
    o.set_officer("First class Officer");
```

```
    cout<<"\n\nAbout Officer: "<<endl;
    o.show();
    regular rt;
    rt.staff_info("Yash",16);
    rt.set_typist(200.5);
    rt.set_regular(2000);
    cout<<"\n\nAbout Regular Typist: "<<endl;
    rt.show();
    casual ct;
    ct.staff_info("Jay",15);
    ct.set_typist(78.5);
    ct.set_casual(4000);
    cout<<"\n\nAbout Casual Typist: "<<endl;
    ct.show();
}
```

Output:

```
C:\Users\Himani\Desktop\PPL\4.exe
Name      Code        Subject          Publication                Highest qualification in general education      Highest professional qualification
Kush       20      Social Studies     How the world has changed                      BE                                      Masters in Geography


About Officer:
Name      Code       Catagory                Highest qualification in general education      Highest professional qualification
Abhishek  14      First class Officer                  B.tech                                 Masters in Structural Engineering


About Regular Typist:
Name             Code          Speed          Wage
Yash              16           200.5           2000


About Casual Typist:
Name             Code          Speed          Wage
Jay               15           78.5            4000


--------------------------------
Process exited after 0.03829 seconds with return value 0
Press any key to continue . . .
```