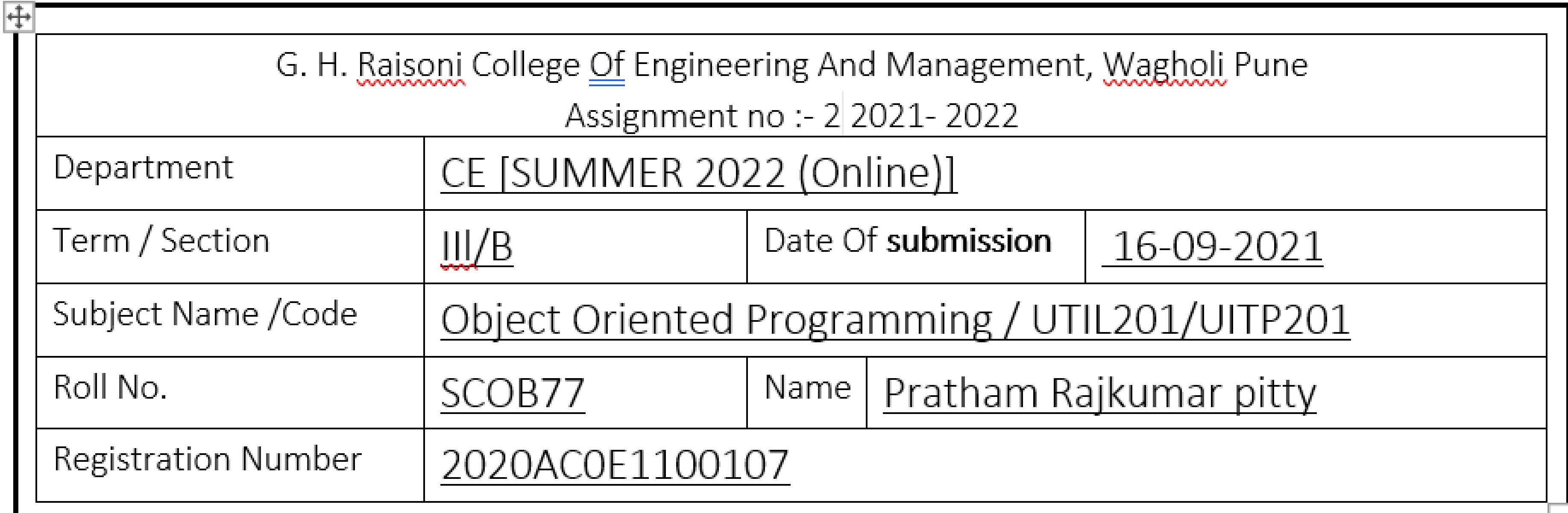


G. H. Raisoni College Of Engineering And Management, Wagholi Pune

Assignment no :- 2 2021- 2022

Department	CE [SUMMER 2022 (Online)]		
Term / Section	III/B	Date Of **submission**	16-09-2021
Subject Name /Code	Object Oriented Programming / UTIL201/UITP201		
Roll No.	SCOB77	Name	Pratham Rajkumar pitty
Registration Number	2020AC0E1100107		

Assignment 2

Aim → Define a class to represent a bank account including the following members:

- Data members → Name of depositor, Account number, Type of account 1, Balance amount in the account
- member Function → To assign initial values To deposit an amount, To withdraw an amount after checking the balance, To display name & balance

write a main program to test program using class and object

Theory →

▸ Object Oriented Programming (OOP)

- OOP is a programming style that is associated with the concept of class, object and variables other conceptes like inheritance, hiding, polymorphism, etc. in ~~the~~ programming.

- The main aim of oop is to bind together the data and function that operate on them so that no other part of the code can access this data except that function.

▸ OOP's Concept

1. objects	3. Abstraction	5. inheritance
2 Class	4. Encapsulation	6. Overloading

7. Exception Handling.

•1 Class :-> →Class is a boilding block in OOP
→ It is a user-defined data type, which holds its own datamembers and member functions. which can be accessed and Used by creating an instance of that class. A class is like a blueprint for an object.

Ex- class_Fruit | object
Mango, orange

•2 object :-> An Object is an instance of a class

when a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.

Eg.

```
class person
{
 char name [20];
  int id;
public:
   void getdetails() {}
 }
int main()
{
 person p1; // p1 is a objet
 }
```

Object take up space in memory and have an associated address like a record in pascal or structure or union in c

• 3 **Abstraction :-**

Abstraction refers to Showing only the essential ~~feach~~ Features of the application and hiding the details in background or implementation.

: Abstraction in class :→ Ab we can implement abstraction in c++ using class .

: Abstraction in Header files :→ Abstraction in C++ can be header files.

Ex. If we have to do some calculation to ~~the~~ find the Power Of any number. we can use the <math.h> header file.

• 4 **Encapsulation :→** def^n :> wrapping up of data and information under a single unit.

in OOP ~~is~~ encapsulation is defined as ~~be~~ binding together the data and the functions that manipulate Them

• 5 **Inheritance :→** The capability of a class to derive properties and characteristics from another class is called inheritance

way to reuse once written code again and again.

: Sub class :→ The class that inherits properties from another class is called Sub Class or Derived class or parent a~~nd child~~ class.

: Super class :→ The class whose properties are inherited by sub class is called the base class.

: Reusability :→ Inheritance supports the concepts of "resability".

•6 Polymorphism :→ means having many forms

It lets us create functions with same name but different arguments, which will perform different actions.
i.e. functions with same name, but functioning in different way.

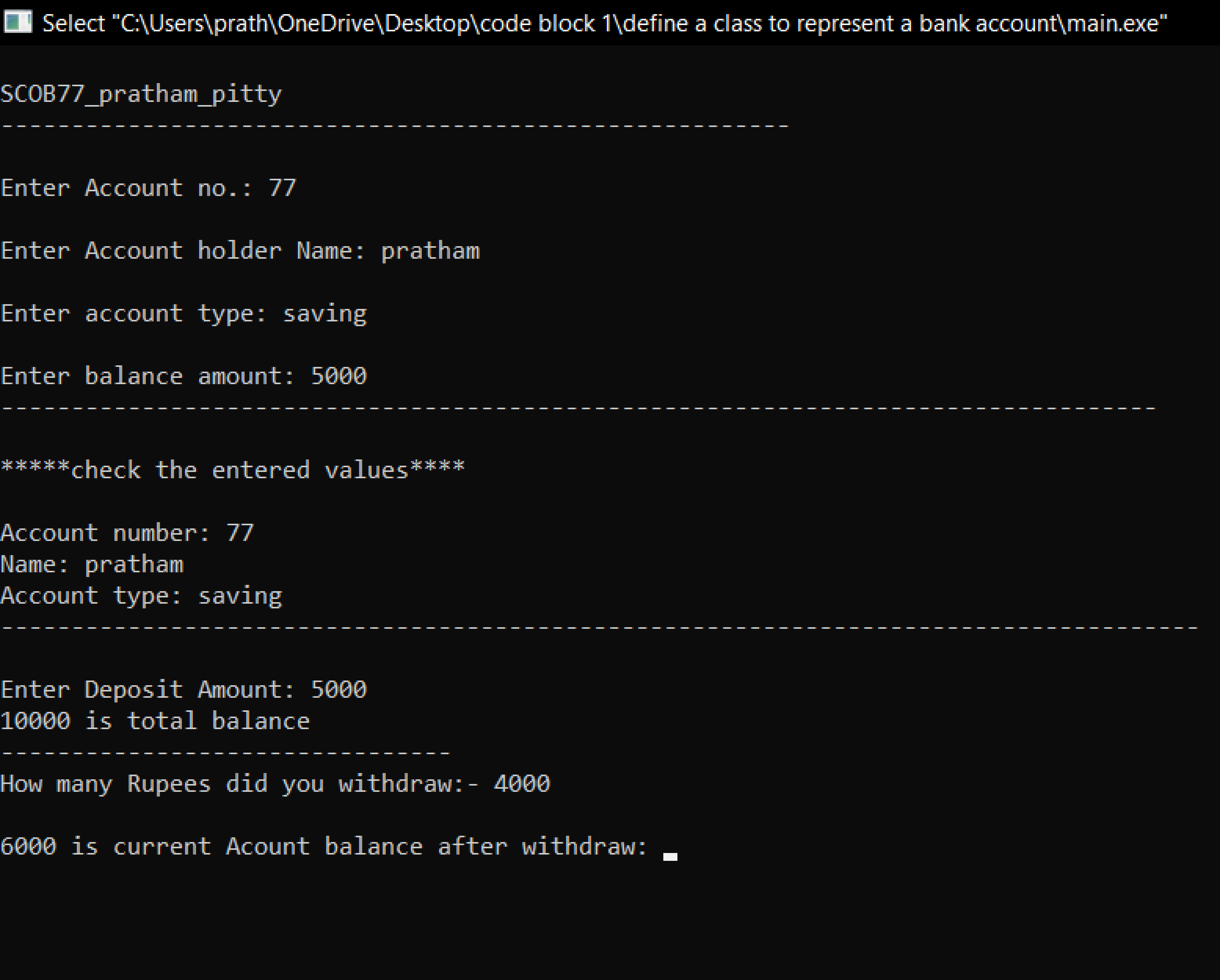
C++ supports operator overloading and function overloading.

: Operator overloading :→ The process of making an operator to exhibit different behaviours in different instances is known as operator overloding

: Function overloading :→ function overloading is using a single function name to perform different types of tasks.

polymorphism is extensively used in implementing inheritance.

```
#include<stdio.h>
#include<iostream>
#include<conio.h>
#include<string.h>
using namespace std;
class bank
{
char name[20];
int Ano;
char atype[20];
float bal;
public:
void get(int no,char *n,char *t,float b)
{
strcpy(name,n);
Ano=no;
strcpy(atype,t);
bal=b;
}
float deposit()
{
float amt;
cin>>amt;
bal=bal+amt;
return bal;
}
float withdrw()
{
float amt;
cout<<"\nHow many Rupees did you withdraw:- ";
cin>>amt;cout<<"\n";
bal=bal-amt;
return bal;
}
void disp()
{
cout<<"\n\n*****check the entered values****";
cout<<"\n\nAccount number: "<<Ano;
cout<<"\nName: "<<name;
cout<<"\nAccount type: "<<atype;
cout<<"\n---------------------------------------------------------------------------------";
cout<<"\n\nEnter Deposit Amount: "<<deposit()<<" is total balance";
cout<<"\n------------------------------"<<withdrw();
cout<<" is current Acount balance after withdraw: ";
}};
int main(void)
{
int n;
char nm[20],t[20];
float a;
bank bk;
cout<<"\nSCOB77_pratham_pitty\n";
cout<<"-----------------------------------------------------\n";
cout<<"\nEnter Account no.: "; cin>>n;
cout<<"\nEnter Account holder Name: "; cin>>nm;
cout<<"\nEnter account type: "; cin>>t;
cout<<"\nEnter balance amount: ";cin>>a;
cout<<"------------------------------------------------------------------------------";
bk.get(n,nm,t,a);
bk.disp();
getch();
}
```


Select "C:\Users\prath\OneDrive\Desktop\code block 1\define a class to represent a bank account\main.exe"

```

SCOB77_pratham_pitty
--------------------------------------------------------

Enter Account no.: 77

Enter Account holder Name: pratham

Enter account type: saving

Enter balance amount: 5000
----------------------------------------------------------------------------------

*****check the entered values****

Account number: 77
Name: pratham
Account type: saving
-------------------------------------------------------------------------------------

Enter Deposit Amount: 5000
10000 is total balance
--------------------------------
How many Rupees did you withdraw:- 4000

6000 is current Acount balance after withdraw: _
```