

OOP LABORATORY 7

Name: **ANIRBAN HAZRA**

Section: **B-12**

Roll : **2005643**

1. Create a class student which stores name, roll number and age of a student. Derive a class test from student class, which stores marks in 5 subjects. Input and display the details of a student.

PROGRAM CODE:-

```
#include<iostream>
using namespace std;

class student
{
protected:
    char name[20];
    int roll,age;
public:
    void getdata ()
    {
        cout << "Enter roll, name , age" << endl;
        cin >> roll >> name >> age;
    }
};

class test:public student
{
protected:
    int sub[5];
public:

    void getmark ()
    {
        cout << "Enter 5 subjects marks : " << endl;
        cin >> sub[0] >> sub[1] >> sub[2] >> sub[3] >> sub[4];
    }

    void details ()
    {
        cout << "\n\nName : " << name << " Roll number : " << roll << endl;
        cout << "Marks in 5 subjects : " << sub[0] << ", " << sub[1] << ", " <<sub[2] <<
            ", " << sub[3] << ", " << sub[4] << endl;
    }
}
```

```

        cout << "\nAge : " << age << endl;
    }
};

int
main ()
{
    test ob;
    ob.getdata ();
    ob.getmark ();
    ob.details ();
}

```

OUTPUT:-

```

Enter roll, name , age
643
Anirban
24
Enter 5 subjects marks :
70 80 90 85 95

Name : Anirban Roll number : 643
Marks in 5 subjects : 70, 80, 90, 85, 95

Age : 24

```

2. Extend the program 1) to derive a class from result from class 'test' which includes member function to calculate total marks and percentage of a student. Input the data for a student and display its total marks and percentage.

PROGRAM CODE:-

```

#include<iostream>
using namespace std;

class student
{
protected:
    char name[20];
    int roll,age;
public:

```

```

void getdata ()
{
    cout << "Enter roll and name and age" << endl;
    cin >> roll >> name >> age;
}

};

class test:public student
{
protected:
    int sub1;
    int sub2;
    int sub3;
    int sub4;
    int sub5;
public:

    void getmark ()
    {
        cout << "Enter 5 subjects marks : " << endl;
        cin >> sub1 >> sub2 >> sub3 >> sub4 >> sub5;
    }
    void details ()
    {
        cout << "\n\nName : " << name << "\nRoll number : " << roll << endl;
        cout << "\nMarks in 5 subjects : " << sub1 << ", " << sub2 << ", " << sub3
        cout << ", " << sub4 << ", " << sub5 << endl;
        cout<<"\nAge : "<<age<<endl;
    }

};

class result:public test
{
    int total;
    float percent;
public:
    void calc ()
    {
        total = sub1 + sub2 + sub3 + sub4 + sub5;
        percent = (total * 100) / 500;

    }
    void display ()
    {
        cout << "Total Marks = " << total << "\nPercentage = " << percent << endl;
    }

};

```

```

int main ()
{
    result ob1;
    ob1.calc ();
    ob1.getdata ();
    ob1.getmark ();
    ob1.details ();
    ob1.calc ();
    ob1.display ();

    return 0;
}

```

OUTPUT:-

```

Enter roll and name and age
643
Anirban
18
Enter 5 subjects marks :
75 85 95 80 90

Name : Anirban
Roll number : 643

Marks in 5 subjects : 75, 85, 95, 80, 90

Age : 18
Total Marks = 425
Percentage = 85

```

3. Extend the program 1) to include a class sports, which stores the marks in sports activity. Derive the result class from the classes 'test' and 'sports'. Calculate the total marks and percentage of a student.

PROGRAM CODE:-

```

#include<iostream>
using namespace std;

class student
{
protected:
    char name[20];

```

```

    int roll,age;
public:
    void getdata ()
    {
        cout << "Enter roll and name and age" << endl;
        cin >> roll >> name >> age;
    }

};

class test:public student
{
protected:
    int sub1;
    int sub2;
    int sub3;
    int sub4;
    int sub5;

public:
    void getmark ()
    {
        cout << "Enter 5 subjects marks : " << endl;
        cin >> sub1 >> sub2 >> sub3 >> sub4 >> sub5;
    }
    void details ()
    {
        cout << "\n\nName : " << name << " Roll number : " << roll << endl;
        cout << "Marks in 5 subjects : " << sub1 << ", " << sub2 << ", " << sub3
            << ", " << sub4 << ", " << sub5 << endl;
        cout<<"\nAge : "<< age;
    }

};

class sports
{
protected:
    int msports;
public:
    void getspo ()
    {
        cout << "Enter marks in sports : ";
        cin >> msports;

    }
};

class result:public sports, public test
{

```

```

    int total;
    float percent;
public:
    void display ()
    {
        cout << "\nMarks in sports = " << msports << endl;
        total = sub1 + sub2 + sub3 + sub4 + sub5 + msports;
        percent = (total * 100) / 600;
        cout << "Total marks : " << total << "\nPercent = " << percent << endl;
    }
};

int
main ()
{
    result ob1;
    ob1.getdata ();
    ob1.getmark ();
    ob1.getspo ();
    ob1.display ();
    ob1.details ();
    ob1.display ();
}

```

OUTPUT:-

```

Enter roll and name and age
643
Anirban
18
Enter 5 subjects marks :
89 90 91 92 93
Enter marks in sports : 95

Marks in sports = 95
Total marks : 550
Percent = 91

Name : Anirban Roll number : 643
Marks in 5 subjects : 89, 90, 91, 92, 93

Age : 18
Marks in sports = 95
Total marks : 550
Percent = 91

```

4. Create a class 'shape'. Derive three classes from it: Circle, Triangle and Rectangle. Include the relevant data members and functions in all the classes. Find the area of each shape and display it.

PROGRAM CODE:-

```
#include<iostream>
#include<conio.h>
using namespace std;
class shape
{
    public:
        virtual void area()=0;
};
class circle: public shape
{
    float r;
    public:
        void area()
        {
            cout<<"Enter radius of the Circle : ";
            cin>>r;
            cout<<"\nArea of the Circle : "<<(2.146*r*r);
            cout<<"\n";
        }
};
class triangle: public shape
{
    int h,b;
    float a;
    public:
        void area()
        {
            cout<<"\nEnter height of the Triangle : ";
            cin>>h;
            cout<<"\nEnter breadth of the Triangle : ";
            cin>>b;
            a=0.5*h*b;
            cout<<"\nArea of the Triangle : "<<a;
            cout<<"\n";
        }
};
class rectangle: public shape
{
    int l,b;
    public:
        void area()
        {
            cout<<"\nEnter length of the Rectangle : ";
```

```

        cin>>l;
        cout<<"\nEnter breadth of the Rectangle : ";
        cin>>b;
        cout<<"\nArea of the Rectangle : "<<l*b;
        cout<<"\n";
    }
};

int main()
{
    circle c;
    c.area();
    triangle t;
    t.area();
    rectangle r;
    r.area();
    getch();
    return(0);
}

```

OUTPUT:-

```

Enter radius of the Circle : 5

Area of the Circle : 53.65

Enter height of the Triangle : 5

Enter breadth of the Triangle : 8

Area of the Triangle : 20

Enter length of the Rectangle : 4

Enter breadth of the Rectangle : 5

Area of the Rectangle : 20

```


5. Create a class which stores employee name, id and salary Derive two classes from 'Employee' class: 'Regular' and 'Part-Time'. The 'Regular' class stores DA, HRA and basic salary. The 'Part-Time' class stores the number of hours and pay per hour. Calculate the salary of a regular employee and a par-time employee.

PROGRAM CODE:-

```
#include <iostream>
```

```
using namespace std;
```

```
class Employee{  
    protected:  
        string name;  
        int id;  
        double salary;
```

```
    public:
```

```
};
```

```
class Regular: public Employee
```

```
{
```

```
    private:
```

```
        double DA;  
        double HRA;  
        double basic_salary;
```

```
    public:
```

```
        Regular(double d, double h, double b)
```

```
{
```

```
    {  
        DA=d;  
        HRA=h;  
        basic_salary=b;
```

```
    }
```

```
    void display()
```

```
    {  
        cout<<"\nSalary of the Regular employee is "<<(DA+HRA+basic_salary);  
    }
```

```
};
```

```
class PartTime: public Employee
```

```
{
```

```
    private:
```

```
        int number_of_hours;  
        double pay_per_hour;
```

```

public:
    PartTime(int n, double p)
    {
        number_of_hours=n;
        pay_per_hour=p;
    }
    void display(){
        cout<<"\nSalary of the part-time employee is " << (number_of_hours * pay_per_hour );
    }
};

int main()
{
    int da, hra , basic;
    cout<<"\nInsert the DA , HRA and Basic Sal for regular employee: ";
    cin>>da>>hra>>basic;
    Regular r(da,hra,basic);
    r.display();

    int hours, pay;
    cout<<"\nInsert the no.of hours and Pay per hour for part time employee: ";
    cin>>hours>>pay;
    PartTime p(hours,pay);
    p.display();

    return 0;
}

```

OUTPUT:-

```

Insert the DA , HRA and Basic Sal for regular employee: 25000 20000 75000

Salary of the Regular employee is 120000
Insert the no.of hours and Pay per hour for part time employee: 8 800

Salary of the part-time employee is 6400

```

6. Create a class which stores account number, customer name and balance. Derive two classes from 'Account' class: 'Savings' and 'Current'. The 'Savings' class stores minimum balance. The 'Current' class stores the over-due amount. Include member functions in the appropriate class to:-

- deposit money
- withdraw [For saving account minimum balance should be checked.]
- [For current account overdue amount should be calculated.]
- display balance

PROGRAM CODE:-

```
#include<stdio.h>
#include<iostream>
#include<conio.h>

using namespace std;

class account
{
    int ac_no;
    char ac_name[20];
protected:
    int balance;
public:
    void getinfo ();
    void display ();

};

void
account::display ()
{
    cout << "\n" << ac_no;
    cout << "\n" << ac_name;
    cout << "\n" << balance;
}

class sav:public account
{
public:
    sav ()
    {
        balance = 0;
    }
    void withdrawl ();
    void deposit ();
};

class current:public account
{
public:
    current ()
    {
        balance = 2000;
    }
    void withdrawl ();
    void deposit ();
};
```

```

void
sav::withdrawl ()
{

    cout << "\n\n ENTER THE AMOUNT U WANT TO WITHDRAWL\n\n\n";
    int amount;
    cin >> amount;
    if (amount > balance)
    {
        cout << " \n\nU DONT HAVE ENOUGH BALANCE TO WITHDRAWL";
        //exit(0);
    }
    else
    {
        if (amount > 5000)
        {
            cout <<
                "U CANT WITHDRAWL THE AMOUNMT MORE THAN 5000/- IN
SAVING ACCOUNT";
        }
        else
        {
            cout << "withdrawled amount is : " << amount;
            cout << "BALANCE IN UR ACCOUNT IS : ";
            balance = balance - amount;
            cout << balance;
        }
    }
    getch ();
}

```

```

void
current::withdrawl ()
{

    cout << "\n\n ENTER THE AMOUNT U WANT TO WITHDRAWL\n\n\n";
    int amount;
    cin >> amount;
    if (amount > balance)
    {
        cout << " \n\nU DONT HAVE ENOUGH BALANCE TO WITHDRAWL";
        // exit(0);
    }
    else
    {
        cout << "withdrawled amount is : " << amount;
        cout << "BALANCE IN UR ACCOUNT IS : ";
        balance = balance - amount;
        cout << balance;
        getch ();
    }
}

```

```

        if (balance < 1000)
        {
            balance = balance - (.01 * (1000 - balance));
        }
        cout << "\nBALANCE AFTER IMPOSING CHAREGE : " << balance;
    }
    getch ();
}

void
sav::deposit ()
{

    cout << "enter the amount u want to deposit";
    int amount;
    cin >> amount;
    balance = balance + amount;
    cout << "\nUR BALANCE IS : " << balance;
    int r;
    r = amount * (.01);
    balance = balance + r;
    cout << "\nUR BALANCE AFTER ADDING INTEREST IS : " << balance;
    getch ();
}

void
current::deposit ()
{

    cout << "enter the amount u want to deposit";
    int amount;
    cin >> amount;
    balance = balance + amount;
    cout << "\nUR BALANCE IS : " << balance;
    getch ();
}

void
account::getinfo ()
{
    cout << "\nenter the name \n :";
    cin >> ac_name;
    cout << "\nenter the account number\n :";
    cin >> ac_no;
}

int main()
{
    sav s;
    current c;

```

```

int ch;

cout << "Enter account type";
cout << "\nPress 1 for Saving";
cout << "\nPress 2 for Current";
cin >> ch;
if (ch == 1)
    s.getinfo ();
else
    c.getinfo ();
cout << "\n\nWHAT DO U WANT TO DO\n\n";
cout <<
"*****",
int c1 = 1;
while (c1 != 4)
{
    cout << "\n1. TO WITHDRAWL ";
    cout << "\n2. TO DEPOSIT ";
    cout << "\n3. TO DISPLAY ";
    cout << "\n4. TO EXIT ";
    cin >> c1;
    switch (c1)
    {
        case 1:
            if (ch == 1)
                s.withdrawl ();
            else
                c.withdrawl ();
            break;
        case 2:
            if (ch == 1)
                s.deposit ();
            else
                c.deposit ();
            break;
        case 3:
            if (ch == 1)
                s.display ();
            else
                c.display ();
            break;

        case 4:
            break;
    }
}

return 0;
}

```

OUTPUT:-

```
Enter account type
Press 1 for Saving
Press 2 for Current1

enter the name
:Dushyant

enter the account number
:34572

WHAT DO U WANT TO DO

*****
1. TO WITHDRAWL
2. TO DEPOSIT
3. TO DISPLAY
4. TO EXIT 2
enter the amount u want to deposit50000

UR BALANCE IS : 50000
UR BALANCE AFTER ADDING INTEREST IS : 50500
1. TO WITHDRAWL
2. TO DEPOSIT
3. TO DISPLAY
4. TO EXIT 3

34572
Dushyant
50500
```