OBJECT ORIENTED PROGRAMMING

LAB 7

• Name :HITU RAJ

• Roll no. :2005025

• Branch :CSE

/\*Q1.Create a class student which stores name\_025, roll\_025 number and age\_025 of a student. Derive a class

test from student class, which stores marks in 5 subjects. Input and display the details of

a student.using Constructor inheritance\*/

#include <iostream>

#include <string.h>

using namespace std;

class student

{

    char name\_025[50];

    int roll\_025, age\_025;

public:

    student(char \*n, int rol, int ag)

    {

        strcpy(name\_025, n);

        roll\_025 = rol;

        age\_025 = ag;

    }

    void displayd()

    {

        cout << "name , roll , age is" << name\_025 <<" " << roll\_025<<" " << age\_025<<"\n";

    }

};

class test : public student

{

    int marks[5];

public:

    test(int \*mark,char \*n, int rol, int age):student(n,rol,age)

    {

        for (int i = 0; i < 5; i++)

        {

            marks[i] = \*mark++;

        }

    }

    void displaym()

    {

        cout << "marks in 5 subject is" <<"\n";

        for (int i = 0; i < 5; i++)

        {

            cout<<marks[i]<<" ";

        }

    }

};

int main()

{

    int roll\_025, age\_025, marks[5];

    char name\_025[100];

    cout << "enter name\_025 roll\_025 age\_025 : ";

    gets(name\_025);

    cin >> roll\_025>>age\_025;

    cout << "enter the marks in 5 subject";

    for (int i = 0; i < 5; i++)

    {

        cin >> marks[i];

    }

    //CREATING OBJECT OF ONLY THE DERIVED CLASS

    test t(marks,name\_025, roll\_025, age\_025);

    cout << "the entered details of student are\n";

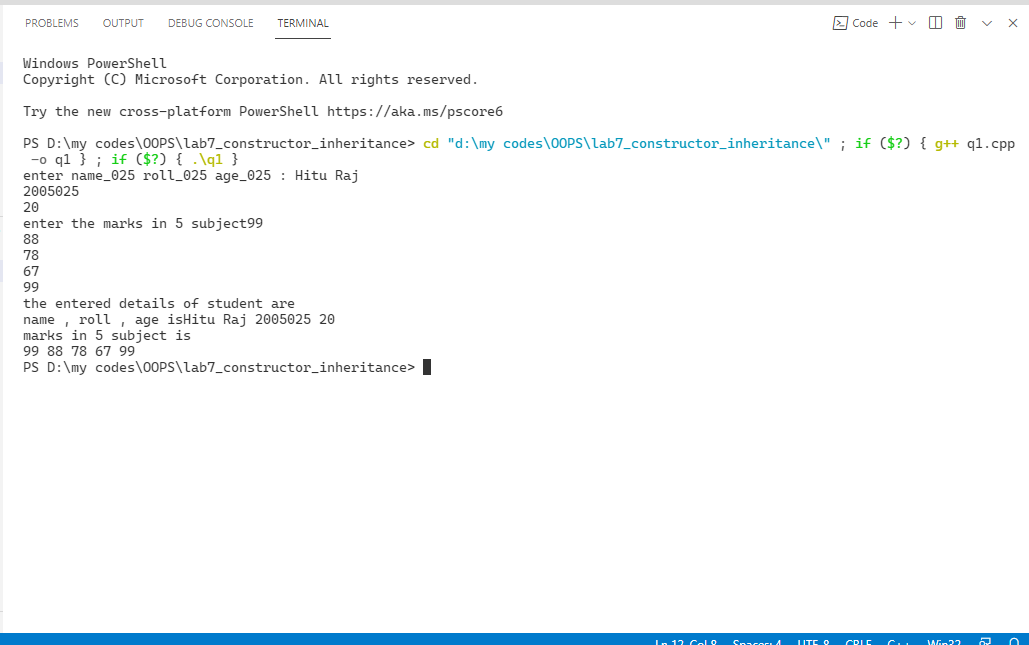
    t.displayd();

    t.displaym();

    return 0;

}

OUTPUT-1



/\*Q2.Extend the program i. 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 his/her grade..\*/

#include <iostream>

#include <string.h>

using namespace std;

class student

{

    char name\_025[50];

    int roll\_025, age\_025;

public:

    student(char \*n, int rol, int ag)

    {

        strcpy(name\_025, n);

        roll\_025 = rol;

        age\_025 = ag;

    }

    void displayd()

    {

        cout << "name\_025 , roll\_025 , age\_025 is" << name\_025 << " " << roll\_025 << " " << age\_025 << "\n";

        cout << "\n.............................................................\n";

    }

};

class test : public student

{

public:

    int marks[5];

    test(int \*mark,char \*n, int rol, int ag):student(n,rol,ag)

    {

        for (int i = 0; i < 5; i++)

        {

            marks[i] = \*mark++;

        }

    }

    void displaym()

    {

        cout << "marks in 5 subject is"

             << "\n";

        for (int i = 0; i < 5; i++)

        {

            cout << marks[i] << " ";

        }

        cout << "\n";

        cout << "\n.............................................................\n";

    }

};

class result : public test

{

    float perc;

    int total\_marks = 0;

    char grade;

public:

  result(int \*mark,char \*n, int rol, int ag):test(mark,n,rol,ag)

  {

  }

    void gradecal()

    {

        for (int i = 0; i < 5; i++)

        {

            total\_marks = total\_marks + marks[i];

        }

        perc = total\_marks / 5.0;

        if (perc >= 90)

        {

            cout << "grade O";

            grade = 'O';

        }

        else if (perc >= 80)

        {

            cout << "grade A";

            grade = 'A';

        }

        else if (perc >= 60)

        {

            cout << "grade B";

            grade = 'B';

        }

        else if (perc >= 40)

        {

            cout << "grade C";

            grade = 'C';

        }

        else

        {

            cout << "you are fail";

            grade = 'F';

        }

    }

};

int main()

{

    int roll\_025, age\_025, marks[5];

    char name\_025[100];

    cout << "enter name\_025 roll\_025 age\_025 : ";

    gets(name\_025);

    cin >> roll\_025 >> age\_025;

    cout << "enter the marks in 5 subject";

    for (int i = 0; i < 5; i++)

    {

        cin >> marks[i];

    }

    //CREATING OBJECT OF ONLY THE DERIVED CLASS

    result t(marks,name\_025, roll\_025, age\_025);

    cout << "the entered details of student are\n";

    t.displayd();

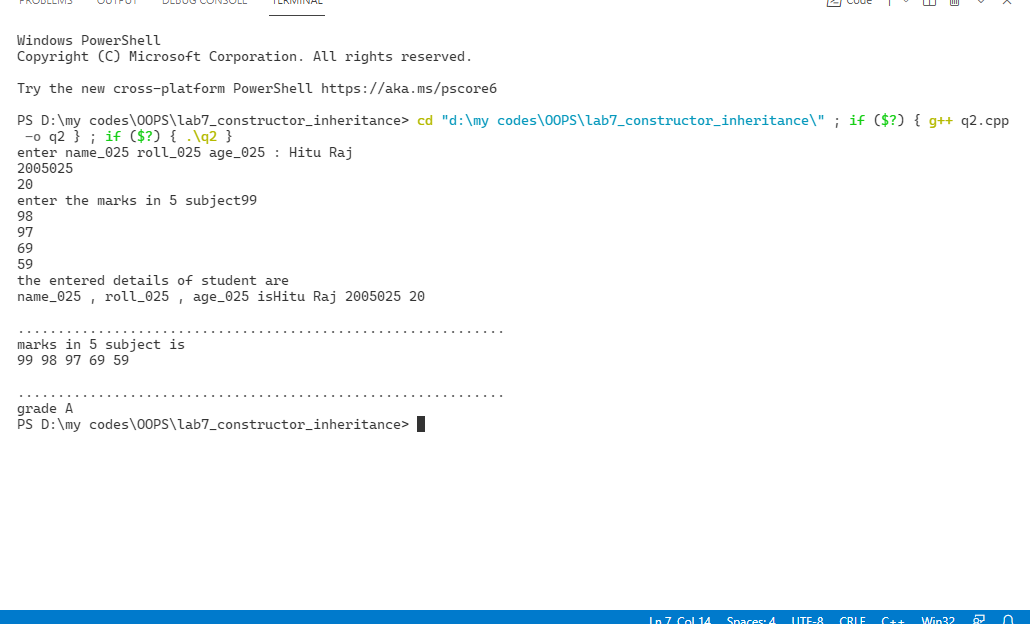
    t.displaym();

    t.gradecal();

    return 0;

}

OUTPUT-2



/\*Q3.Extend the program ii. to include a class sports, which stores the marks\_025 in sports activity. Derive the result class from the classes ‘test’ and ‘Activities’. Calculate the total marks\_025

and percentage of a student.\*/

#include <iostream>

#include <string.h>

using namespace std;

class student

{

    char name\_025[50];

    int roll\_025, age\_025;

public:

    student(char \*n, int rol, int ag)

    {

        strcpy(name\_025, n);

        roll\_025 = rol;

        age\_025 = ag;

    }

    void displayd()

    {

        cout << "name\_025 , roll\_025 , age\_025 is" << name\_025 << " " << roll\_025 << " " << age\_025 << "\n";

        cout << "\n.............................................................\n";

    }

};

class test : virtual public student

{

public:

    int marks\_025[3];

    test(int \*mark, char \*n, int rol, int ag) : student(n, rol, ag)

    {

        for (int i = 0; i < 3; i++)

        {

            marks\_025[i] = \*mark++;

        }

    }

    void displaym()

    {

        cout << "marks\_025 in 3 subject is"

             << "\n";

        for (int i = 0; i < 3; i++)

        {

            cout << marks\_025[i] << " ";

        }

        cout << "\n";

        cout << "\n.............................................................\n";

    }

};

class sport : virtual public student

{

public:

    int activity[2];

    sport(int \*activ, char \*n, int rol, int ag) : student(n, rol, ag)

    {

        for (int i = 0; i < 3; i++)

        {

            activity[i] = \*activ++;

        }

    }

    void displaya()

    {

        cout << "marks\_025 in Activity  is"

             << "\n";

        for (int i = 0; i < 2; i++)

        {

            cout << activity[i] << " ";

        }

        cout << "\n";

        cout << "\n.............................................................\n";

    }

};

class result : public test, public sport

{

    float perc;

    int total\_marks = 0;

    char grade;

public:

    result(int \*mark, int \*activ, char \*n, int rol, int ag) : test(mark, n, rol, ag) , sport(activ, n, rol, ag),student(n,rol,ag)

    {

    }

    void gradecal()

    {

        for (int i = 0; i < 3; i++)

        {

            total\_marks = total\_marks + marks\_025[i];

        }

        for (int i = 0; i < 2; i++)

        {

            total\_marks = total\_marks + activity[i];

        }

        perc = total\_marks / 5.0;

        if (perc >= 90)

        {

            cout << "grade O";

            grade = 'O';

        }

        else if (perc >= 80)

        {

            cout << "grade A";

            grade = 'A';

        }

        else if (perc >= 60)

        {

            cout << "grade B";

            grade = 'B';

        }

        else if (perc >= 40)

        {

            cout << "grade C";

            grade = 'C';

        }

        else

        {

            cout << "you are fail";

            grade = 'F';

        }

    }

};

int main()

{

    int roll\_025, age\_025, marks\_025[3], activ\_marks[2];

    char name\_025[100];

    cout << "enter name\_025 roll\_025 age\_025 : ";

    gets(name\_025);

    cin >> roll\_025 >> age\_025;

    cout << "enter the marks\_025 in 3 subject";

    for (int i = 0; i < 3; i++)

    {

        cin >> marks\_025[i];

    }

    cout << "enter the marks\_025 in 2 Activity ";

    for (int i = 0; i < 2; i++)

    {

        cin >> activ\_marks[i];

    }

    //CREATING OBJECT OF ONLY THE DERIVED CLASS

    result t(marks\_025, activ\_marks, name\_025, roll\_025, age\_025);

    cout << "the entered details of student are\n";

    t.displayd();

    t.displaym();

    t.displaya();

    t.gradecal();

    return 0;

}

OUTPUT-3



/\*Q4.Create a\_025 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.\*/

#include <iostream>

#include <string.h>

#include <math.h>

using namespace std;

class area

{

public:

    void acircle(int r\_025)

    {

        cout << "area of circle is " << 3.14 \* r\_025 \* r\_025 << "squnit";

    }

    void arectangle(int l\_025, int b\_025)

    {

        cout << "area of rectangle is " << l\_025 \* b\_025 << "squnit";

    }

    void atriangle(int a\_025, int b\_025, int c\_025)

    {

        float s = (a\_025 + b\_025 + c\_025) / 2.0;

        float area = pow((s \* (s - a\_025) \* (s - b\_025) \* (s - c\_025)), 1.0 / 2);

        cout << "area of triangle is " << area << " squnit";

    }

};

class circle : public area

{

public:

    int r\_025;

    circle(int r\_025)

    {

        r\_025 = r\_025;

    }

};

class rectangle : public area

{

    int l\_025, b\_025;

public:

    int r\_025;

    rectangle(int l1\_025, int b1\_025)

    {

        l\_025 = l1\_025;

        b\_025 = b1\_025;

    }

};

class triangle : public area

{

public:

    int a\_025,b\_025,c\_025;

    triangle(int a1,int b1,int c1)

    {

       a\_025=a1;

       b\_025=b1;

       c\_025=c1;

    }

};

int main()

{

    int n;

    cout << "Press 1 to calculate the area of circle\n";

    cout << "Press 2 to calculate the area of rectangle\n";

    cout << "Press 3 to calculate the area of traingle\n";

    cin >> n;

    switch (n)

    {

    case 1:

    {

        int r\_025;

        cout << "enter the radius";

        cin >> r\_025;

        circle t(r\_025);

        t.acircle(r\_025);

        break;

    }

    case 2:

    {

        int l\_025, b\_025;

        cout << "enter the length and breadth";

        cin >>l\_025>>b\_025;

        rectangle t(l\_025,b\_025);

        t.arectangle(l\_025,b\_025);

        break;

    }

    case 3:

    {

        int a\_025, b\_025, c\_025;

        cout << "enter the  3 sides";

        cin >> a\_025 >> b\_025 >> c\_025;

        triangle t(a\_025, b\_025, c\_025);

        t.atriangle(a\_025, b\_025, c\_025);

        break;

    }

    default:

    cout<<"\nWrong Imtput";

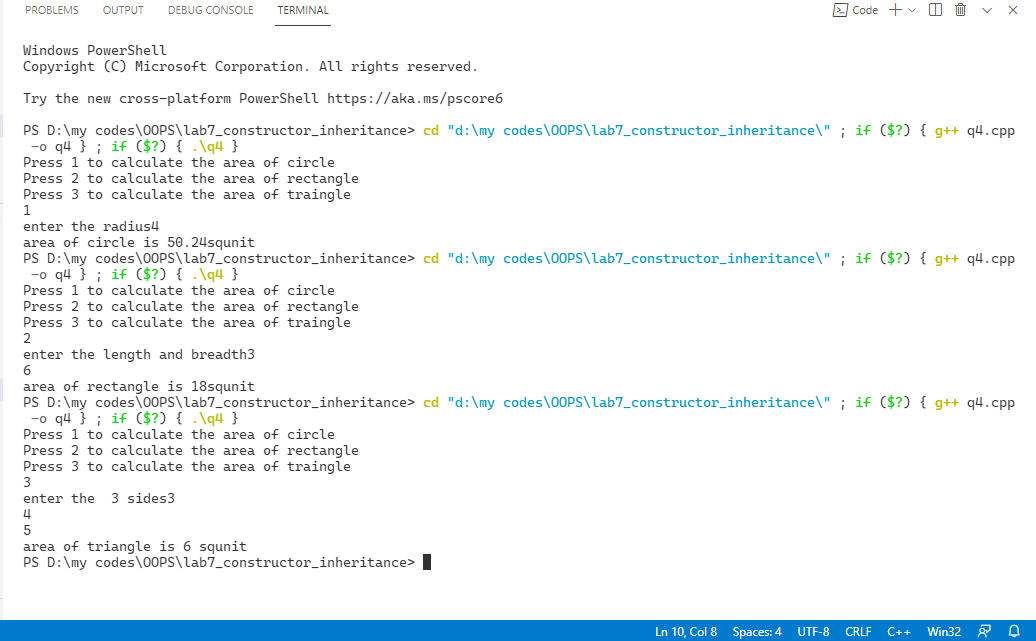
        break;

    }

    return 0;

}

OUTPUT-4



/\*Q5.Create a class which stores employee name\_025,id\_025 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\_025 and pay\*/

#include <iostream>

#include <string.h>

using namespace std;

class employee

{

    char name\_025[20];

    int id\_025;

public:

    employee(char \*nam, int ide)

    {

        strcpy(name\_025, nam);

        id\_025 = ide;

    }

    void display()

    {

        cout << "\nname of employee: " << name\_025;

        cout << "\nid of employee: " << id\_025;

    }

};

class regular : public employee

{

    int basic\_sal;

    float da\_025, hra\_025, sal\_025;

public:

    regular(int basic\_sal1, char \*nam, int ide) : employee(nam, ide)

    {

        basic\_sal = basic\_sal1;

        da\_025 = 0.8 \* basic\_sal;

        hra\_025 = 0.1 \* basic\_sal;

        sal\_025 = basic\_sal + da\_025 + hra\_025;

    }

    void displayem()

    {

        display();

        cout << "the total salary of the employee is " << sal\_025;

    }

};

class part\_time : public employee

{

    int hours\_025, pay\_per\_hour, sal\_025;

public:

    part\_time(int hr, int pph, char \*lora, int id\_025) : employee(lora, id\_025)

    {

        hours\_025 = hr;

        pph = pay\_per\_hour;

        sal\_025 = hours\_025 \* pay\_per\_hour;

    }

    void displaypt()

    {

        display();

        cout << "\nsalary per hours\_025: " << sal\_025;

    }

};

int main()

{

    int basic\_sal, hours\_025, pay\_per\_hour, n, id\_025;

    char name\_025[100];

    cout << "press1 if u have employee\n";

    cout << "press 2 if u have partime employee\n ";

    cin >> n;

    switch (n)

    {

    case 1:{

        cout << "\nfor regular employees: ";

        cout << "\nenter the name\_025 and id\_025 of employee respectively: ";

        cin >> name\_025 >> id\_025;

        cout << "\nenter the basic salary of employee: ";

        cin >> basic\_sal;

        regular r(basic\_sal, name\_025, id\_025);

        r.displayem();

        break;

    }

    case 2:{

        cout << "\nfor part-time employees: ";

        cout << "\nenter the name\_025 and id\_025 of employee respectively: ";

        cin >> name\_025 >> id\_025;

        cout << "\nenter the total no. of working hours\_025 of emmployee and pay per hour:";

        cin >> hours\_025 >> pay\_per\_hour;

        part\_time p(hours\_025, pay\_per\_hour, name\_025, id\_025);

        p.displaypt();

        break;

    }

    default:

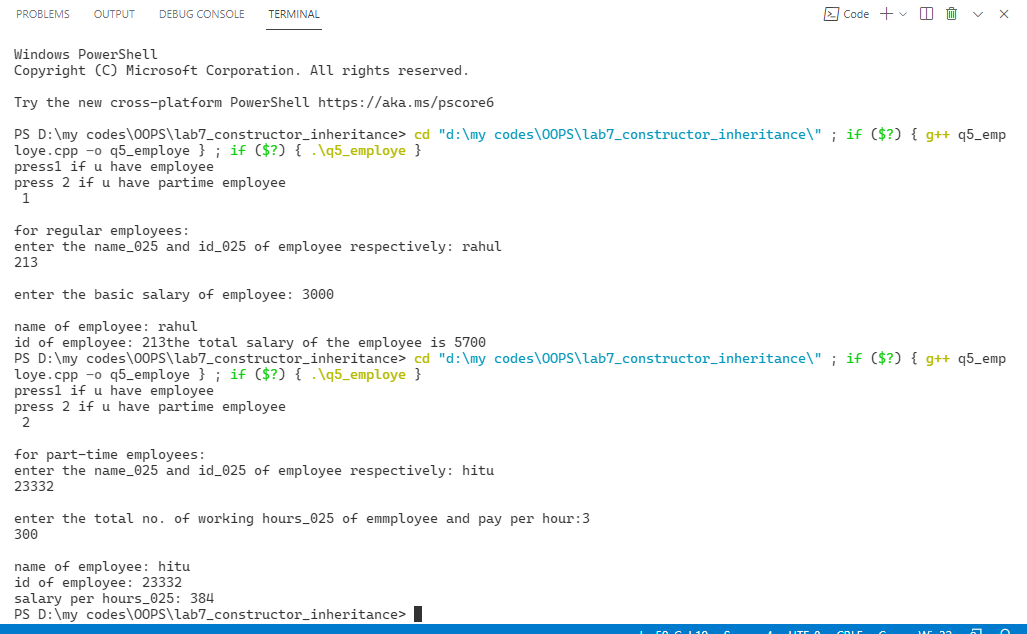
        break;

    }

    return 0;

}

OUTPUT-5



/\*Q6. Cricketer problem(Hierarchial inheritance)\*/

#include <iostream>

#include <string.h>

using namespace std;

class cricketer

{

protected:

    char name[20];

    int age;

    int jn;

    int odi;

public:

    cricketer(char \*name1, int age1, int jn1, int odi1)

    {

        strcpy(name, name1);

        age = age1;

        jn = jn1;

        odi = odi1;

    }

    void display()

    {

        cout << "\nname of cricketer: " << name;

        cout << "\nage of cricketer: " << age;

        cout << "\njersey no. of cricketer: " << jn;

        cout << "\nno. of ODIs played by cricketer: " << odi;

    }

};

class batsman : public cricketer

{

protected:

    int rt, hc, c, hs, ns;

public:

    batsman(int rt1, int hc1, int c1, int hs1, int ns1, char \*name1, int age1, int jn1, int odi1) : cricketer(name1, age1, jn1, odi1)

    {

        rt = rt1;

        hc = hc1;

        c = c1;

        hs = hs1;

        ns = ns1;

    }

    void display()

    {

        cricketer::display();

        cout << "\nrunstaken: " << rt;

        cout << "\nno. of half centuries: " << hc;

        cout << "\nno. of centuries: " << c;

        cout << "\nhighest score: " << hs;

        cout << "\ntotal no. of sixes: " << ns;

    }

};

class baller : public cricketer

{

protected:

    char type[5];

    int wt, s;

public:

    baller(char \*type1, int wt1, int s1, char \*name1, int age1, int jn1, int odi1) : cricketer(name1, age1, jn1, odi1)

    {

        strcpy(type, type1);

        wt = wt1;

        s = s1;

    }

    void display()

    {

        cricketer::display();

        cout << "\nballer type: " << type;

        cout << "\ntotal no. of wickets taken: " << wt;

        cout << "\nspeed of baller: " << s << "km/h";

    }

};

int main()

{

    int age, jn, odi, rt, hc, c, hs, ns, wt, s, n;

    char name[50], type[10];

    cout << "\npress 1 if you have batsman";

    cout << "\npress 2 if you have baller";

    cin >> n;

    switch (n)

    {

    case 1:

    {

        cout << "\nfor batsman: ";

        cout << "\nenter the cricketer's name,age,jersy number and no. of ODIs played respectively: ";

        cin >> name >> age >> jn >> odi;

        cout << "\nenter the runstaken,no. of half centuries and centuries,highest score, no.of sixes : ";

        cin >> rt >> hc >> c >> hs >> ns;

        batsman bt(rt, hc, c, hs, ns, name, age, jn, odi);

        bt.display();

    break;

    }

    case 2:

    {

        cout << "\n\nfor baller: ";

        cout << "\nenter the cricketer's name,age,jersy number and no. of ODIs played respectively: ";

        cin >> name >> age >> jn >> odi;

        cout << "\nenter the type of baller,total no. of wickets taken and speed of baller:";

        cin >> type >> wt >> s;

        baller bl(type, wt, s, name, age, jn, odi);

        bl.display();

    }

default:

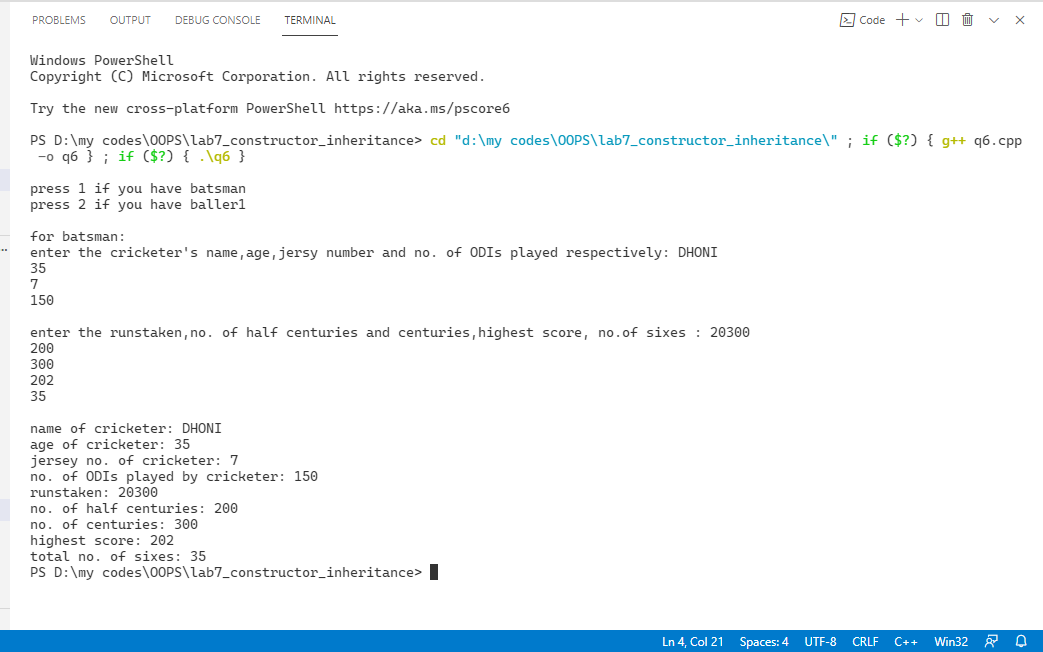
    break;

}

return 0;

}

OUTPUT-6



/\*Q7. Cricketer problem(Multiple  inheritance)\*/

#include <iostream>

#include<string.h>

using namespace std;

class batsman

{

    int rt\_025, hc\_025, c\_025, hs\_025, ns\_025;

public:

     batsman(int rt1, int hc1, int c1, int hs1, int ns1)

    {

        rt\_025 = rt1;

        hc\_025 = hc1;

        c\_025 = c1;

        hs\_025 = hs1;

        ns\_025 = ns1;

    }

    void display\_bt()

    {

        cout << "\nrunstaken: " << rt\_025;

        cout << "\nno. of half centuries: " << hc\_025;

        cout << "\nno. of centuries: " << c\_025;

        cout << "\nhighest score: " << hs\_025;

        cout << "\ntotal no. of sixes: " << ns\_025;

    }

};

class baller

{

    char type\_025[5];

    int wt\_025, s\_025;

public:

   baller(char \*type1, int wt1, int s1)

    {

        strcpy(type\_025, type1);

        wt\_025 = wt1;

        s\_025 = s1;

    }

    void display\_bl()

    {

        cout << "\nballer type\_025: " << type\_025;

        cout << "\ntotal no. of wickets taken: " << wt\_025;

        cout << "\nspeed of baller: " << s\_025 << "km/h";

    }

};

class allrounder : public batsman, public baller

{

    int mm\_025, im\_025, cat\_025, icc\_025;

public:

    allrounder(int mm1,int im1,int cat1, int icc1,int rt1, int hc1, int c1, int hs1, int ns1,char \*type1, int wt1, int s1):baller(type1,wt1, s1),batsman(rt1, hc1, c1, hs1, ns1)

    {

       mm\_025=mm1;

       im\_025=im1;

       cat\_025=cat1;

       icc\_025=icc1;

    }

    void display\_all()

    {

        cout << "\nno. of man of the match: " << mm\_025;

        cout << "\nno. of international matches: " << im\_025;

        cout << "\nno. of catches: " << cat\_025;

        cout << "\nICC ranking: " << icc\_025;

    }

};

int main()

{

     int age\_025, jn, odi, rt\_025, hc\_025, c\_025, hs\_025, ns\_025, wt\_025, s\_025, n,mm\_025,im\_025,cat\_025,icc\_025;

    char name[50], type\_025[10];

     cout<<"\n..................................................\n";

        cout<<"For batsman enter\n";

    cout << "\nenter the runstaken,no. of half centuries and centuries,highest score, no.of sixes : ";

        cin >> rt\_025 >> hc\_025 >> c\_025 >> hs\_025 >> ns\_025;

         cout<<"\n..................................................\n";

        cout<<"For baller enter\n";

        cout << "\nenter the type\_025 of baller,total no. of wickets taken and speed of baller:";

        cin >> type\_025 >> wt\_025 >> s\_025;

        cout<<"\n..................................................\n";

        cout<<"For all rounder enter\n";

        cout << "\nenter the no. of man of the match,no. of international matches,no.of catches and ICC ranking: ";

        cin >> mm\_025 >> im\_025 >> cat\_025 >> icc\_025;

        allrounder a(mm\_025,im\_025,cat\_025,icc\_025,rt\_025, hc\_025, c\_025, hs\_025, ns\_025,type\_025, wt\_025, s\_025);

          a.display\_bt();

        a.display\_bl();

        a.display\_all();

    a.display\_all();

    return 0;

}

OUTPUT-7

