KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY (KIIT)

Deemed to be University U/S 3 of UGC Act, 1956

OBJECT ORIENTED PROGRAMMING

LAB 7

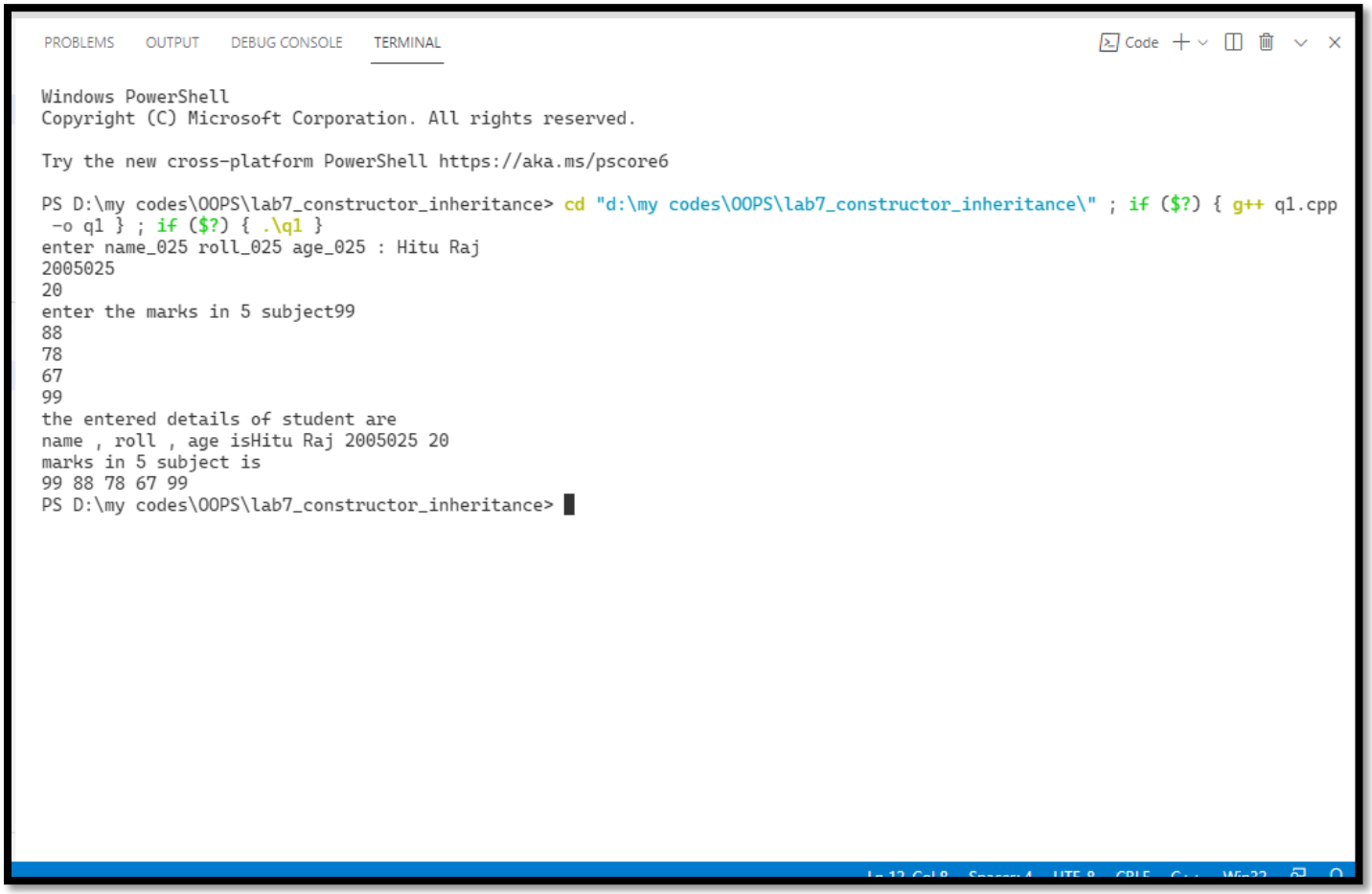
- Name :HITU RAJ
- Roll no. :2005025
- Branch :CSE


```
/*Q1.Create a class student which stores name_025, roll_025 number a
nd 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


```cpp
/*Q3.Extend the program ii. to include a class sports, which stores
the marks_025 in sports activity. Derive the result class from the c
lasses ‘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(m
ark, 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

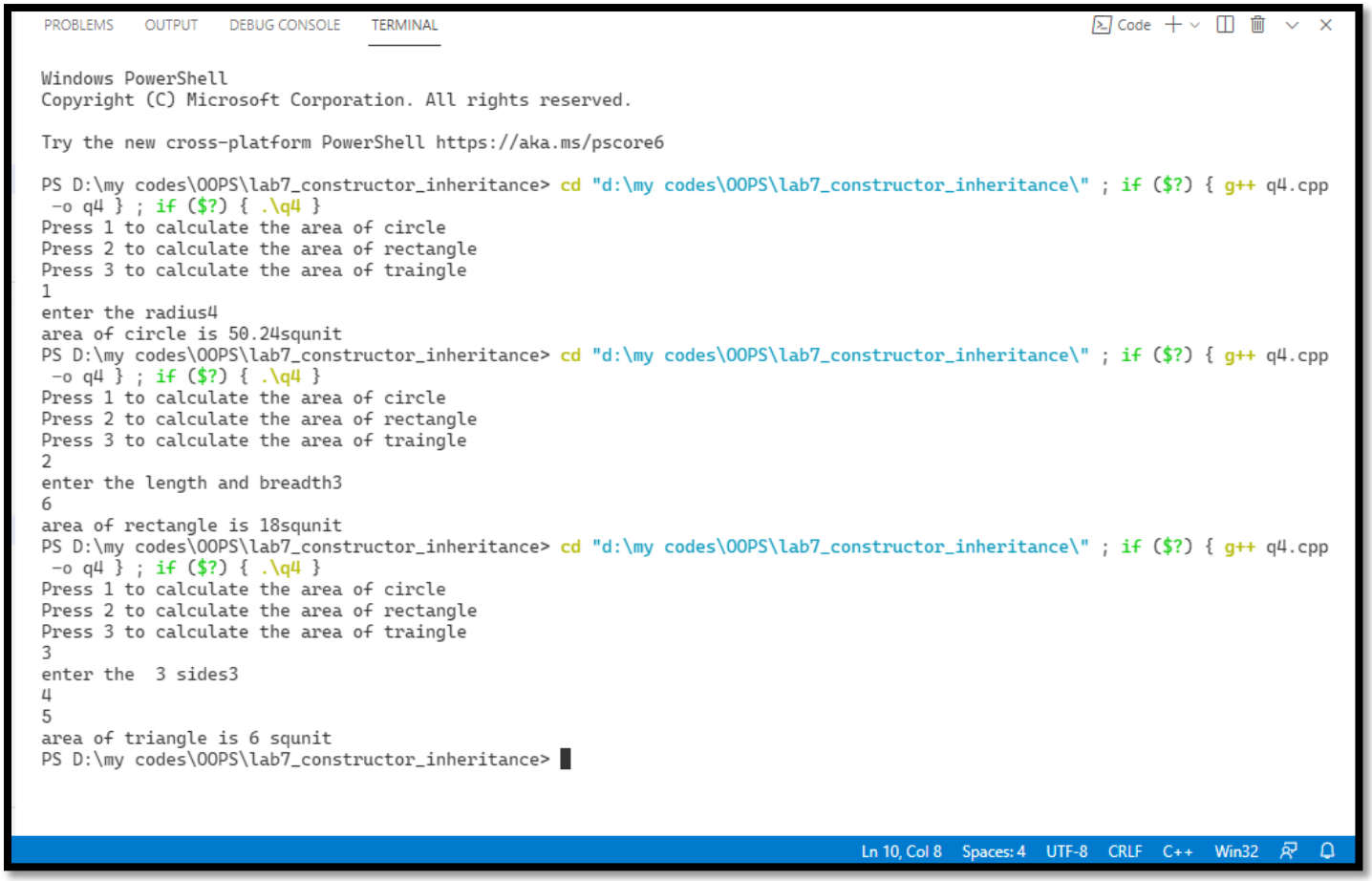
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL                                  Code

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\my codes\OOPS\lab7_constructor_inheritance> cd "d:\my codes\OOPS\lab7_constructor_inheritance\" ; if ($?) { g++ q3.cpp
 -o q3 } ; if ($?) { .\q3 }
enter name_025 roll_025 age_025 : Hitu Raj
2005025
20
enter the marks_025 in 3 subject98
97
96
enter the marks_025 in 2 Activity 99
94
the entered details of student are
name_025 , roll_025 , age_025 isHitu Raj 2005025 20

.............................................................
marks_025 in 3 subject is
98 97 96

.............................................................
marks_025 in Activity  is
99 94

.............................................................
grade O
PS D:\my codes\OOPS\lab7_constructor_inheritance>

Ln 8, Col 2  Spaces: 4  UTF-8  CRLF  C++  Win32
```

```
/*Q4.Create a_025 class ‘shape’. Derive three classes from it: Circl
e, Triangle and Rectangle. Include the relevant data members and fun
ctions 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 << "squ
nit";
    }
    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



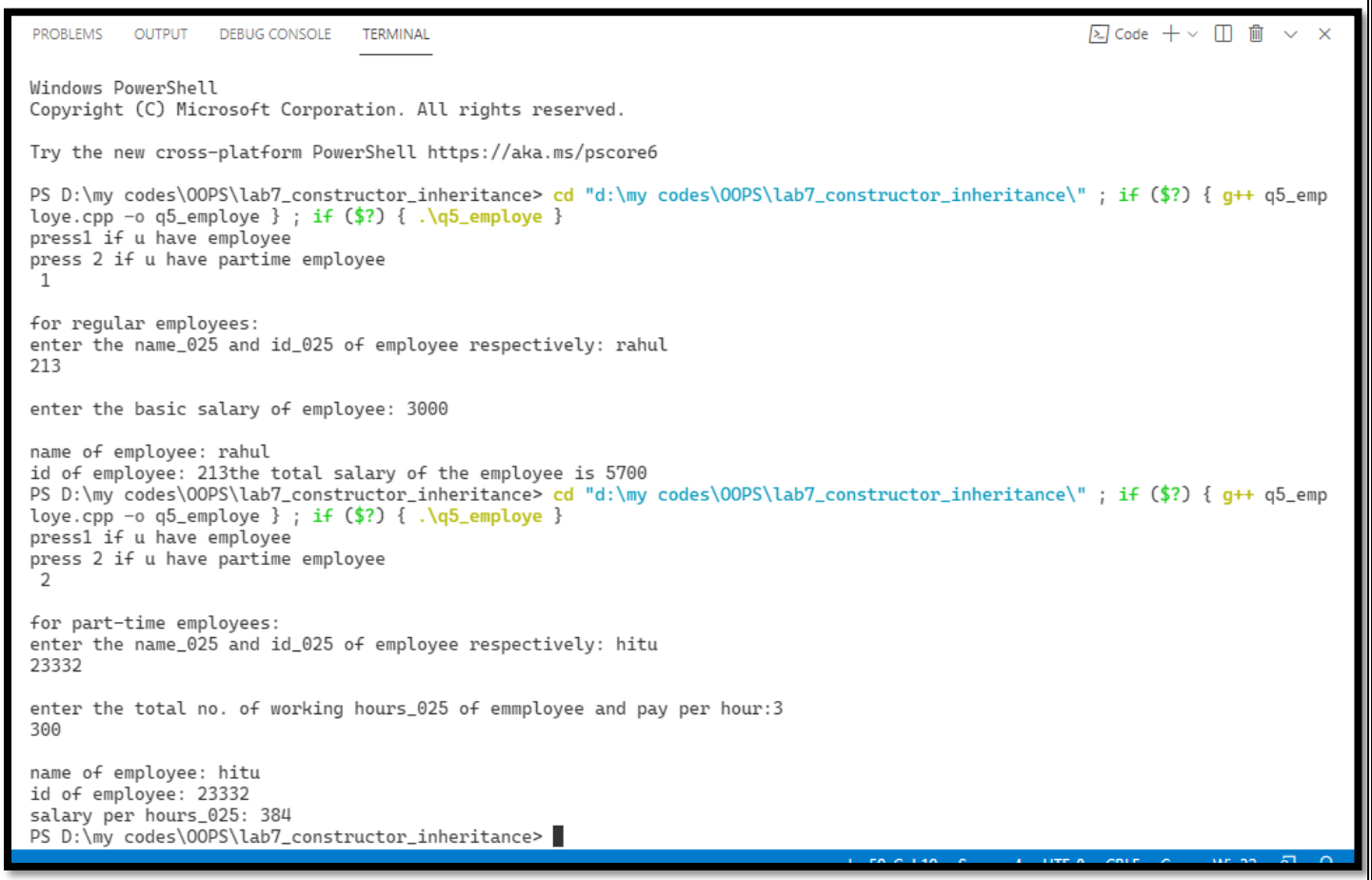

```
/*Q5.Create a class which stores employee name_025,id_025 and salary
 Derive two classes from
‘Employee’ class: ‘Regular’ and ‘Part-
Time’. The ‘Regular’ class stores DA, HRA and
basic salary. The ‘Part-
Time’ class stores the number of hours_025 and pay*/
#include <iostream>
#include <string.h>
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(lo
ra, 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 respect
ively: ";
        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 respect
ively: ";
        cin >> name_025 >> id_025;

        cout << "\nenter the total no. of working hours_025 of emmpl
oyee 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



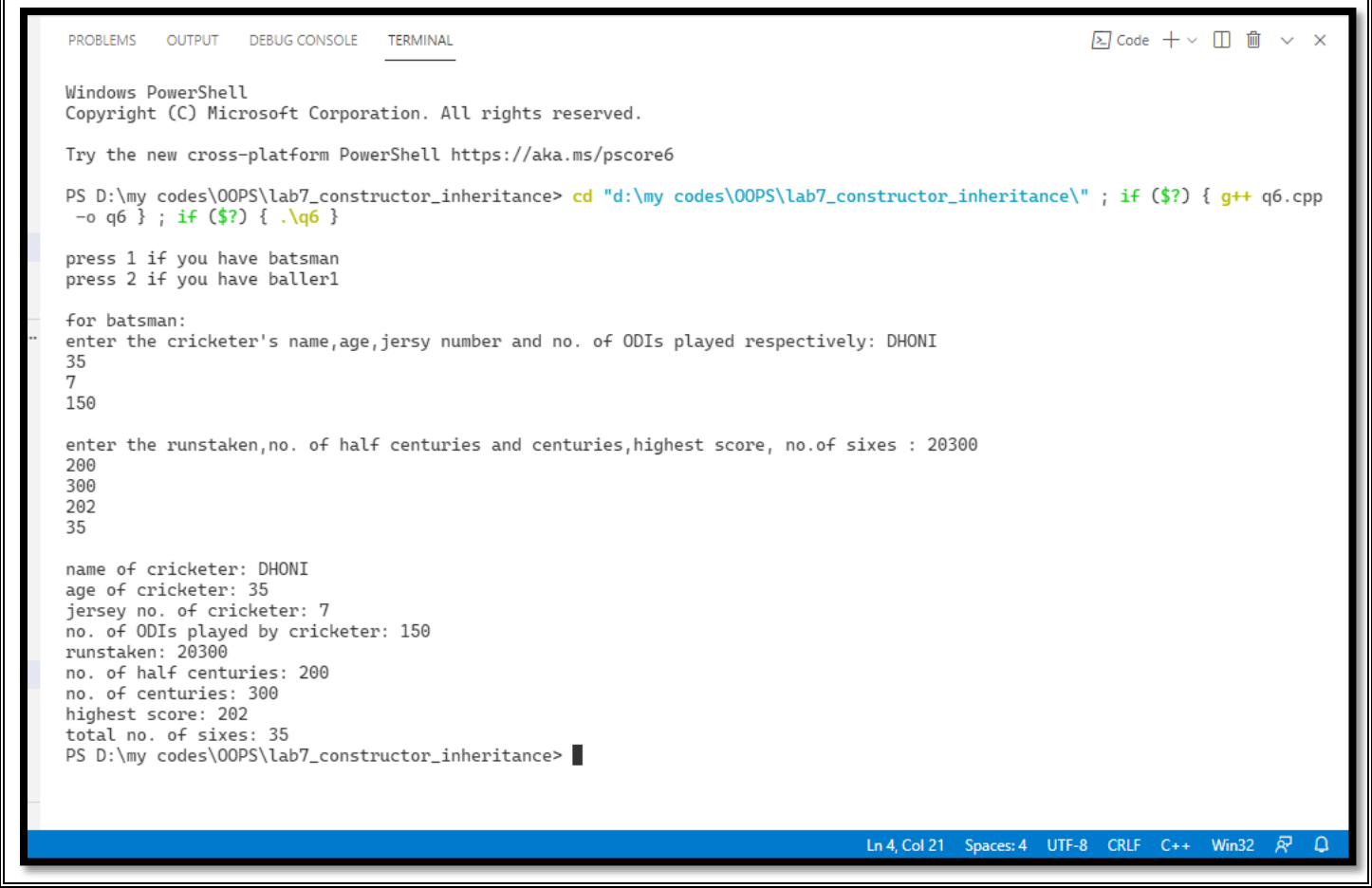

```cpp
/*Q6. Cricketer problem(Hierarchial inheritance)*/
#include <iostream>
#include <string.h>
using namespace std;
class cricketer
{
protected:
    char name[20];
    int age;
    int jn;
    int odi;

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 n
o. of ODIs played respectively: ";
        cin >> name >> age >> jn >> odi;

        cout << "\nenter the runstaken,no. of half centuries and cen
turies,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 n
o. of ODIs played respectively: ";
        cin >> name >> age >> jn >> odi;

        cout << "\nenter the type of baller,total no. of wickets tak
en 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



```cpp
/*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 centuri
es,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 internat
ional 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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\my codes\OOPS\lab7_constructor_inheritance> cd "d:\my codes\OOPS\lab7_constructor_inheritance\" ; if ($?) { g++ q7.cpp
 -o q7 } ; if ($?) { .\q7 }

................................................
For batsman enter

enter the runstaken,no. of half centuries and centuries,highest score, no.of sixes : 200
34
22
333
3

................................................
For baller enter

enter the type_025 of baller,total no. of wickets taken and speed of baller:dsdd
32
32322

................................................
For all rounder enter

enter the no. of man of the match,no. of international matches,no.of catches and ICC ranking: 322
32
32
322

runstaken: 200
no. of half centuries: 34
no. of centuries: 22
highest score: 333
total no. of sixes: 3
baller type_025: dsdd
total no. of wickets taken: 32
speed of baller: 32322km/h
no. of man of the match: 322
no. of international matches: 32
no. of catches: 32
ICC ranking: 322
no. of man of the match: 322
no. of international matches: 32
no. of catches: 32
ICC ranking: 322
PS D:\my codes\OOPS\lab7_constructor_inheritance>
```

Ln 72, Col 5 Spaces: 4 UTF-8 CRLF C++ Win32

2005025_Hitu raj