**LAB ASSIGNMENT-3**

// Q1. WAP to find area of a\_025 circle, a\_025 rectangle and a\_025 triangle, using concept of

//function overloading.

#include <iostream>

#include <math.h>

using namespace std;

void area(double r\_025)

{

  double area = 3.14 \* r\_025 \* r\_025;

  cout << "\narea of circle is " << area << "sq.unit\n";

}

void area(double l\_025, double b\_025)

{

  double area = l\_025 \* b\_025;

  cout << "\narea of rect is " << area << "sq.unit\n";

}

void area(double a\_025, double b\_025, double c\_025)

{

  double s = (a\_025 + b\_025 + c\_025) / 2;

  double area = sqrt(s \* (s - a\_025) \* (s - b\_025) \* (s - c\_025));

  cout << "\narea of triangle is " << area << "sq.unit\n";

}

int main()

{

  double r\_025, a\_025, b\_025, c\_025, l\_025,br\_025;

  int n\_025;

  cout << "\npress 1 to calculate the area of circle"

       << "\npress 2 to calculate the area of reactangle"

       << "\npress 3 to calculate the area of triangle";

  cin >> n\_025;

  switch (n\_025)

  {

  case 1:

    cout << "enter the value of radius";

    cin >> r\_025;

    area(r\_025);

    break;

  case 2:

    cout << "enter the length and breath";

    cin >> l\_025 >>br\_025;

    area(l\_025,br\_025);

    break;

  case 3:

    cout << "enter the length of the 3 sides";

    cin>>a\_025>>b\_025>>c\_025;

    area(a\_025, b\_025, c\_025);

  default:

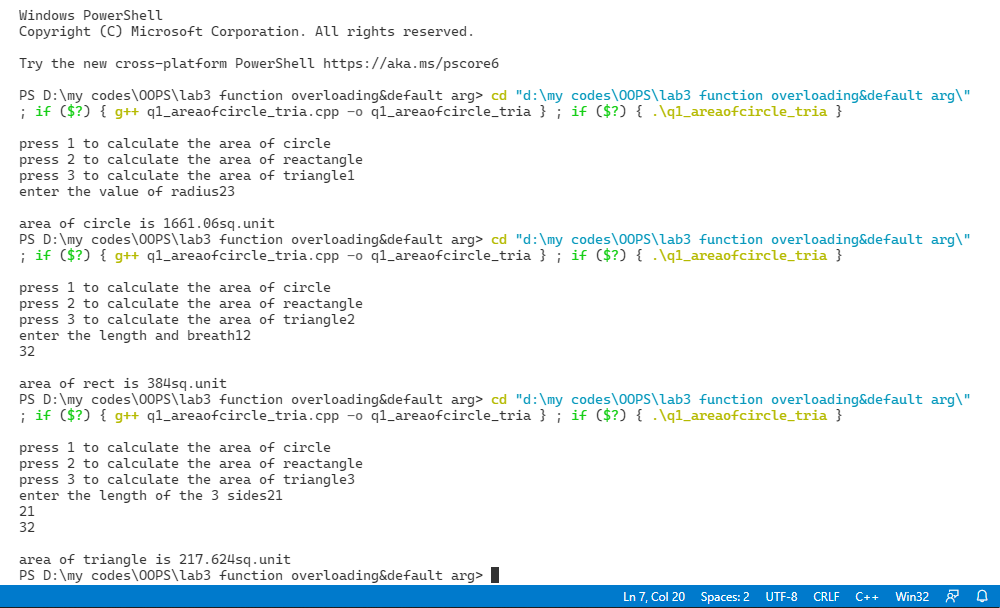
    break;

  }

  return 0;

}

OUTPUT-Q1



//Q2.WAP to find volume of a\_025 sphere, a\_025 cylinder and a\_025 cuboid, using function

//overloading.

#include <iostream>

#include <math.h>

using namespace std;

void volume(double r\_025)

{

   double volume = 4 / 3.0 \* 3.14 \* r\_025 \* r\_025 \* r\_025;

   cout << "\nvolume of sphere is " << volume << " sq.unit\n";

}

void volume(double r\_025, double h\_025)

{

   double volume = 3.14 \* r\_025 \* r\_025 \* h\_025;

   cout << "\nvolume of cylinder is " << volume << " sq.unit\n";

}

void volume(double a\_025, double b\_025, double c\_025)

{

   double s = (a\_025 + b\_025 + c\_025) / 2;

   double volume = sqrt(s \* (s - a\_025) \* (s - b\_025) \* (s - c\_025));

   cout << "\nvolume of triangle is " << volume << " sq.unit\n";

}

int main()

{

   double r\_025, a\_025, b\_025, c\_025, h\_025;

   int n\_025;

   cout << "\npress 1 to calculate the volume  of sphere"

        << "\npress 2 to calculate the volume  of cylinder"

        << "\npress 3 to calculate the volume  of cuboid\n";

   cin >> n\_025;

   switch (n\_025)

   {

   case 1:

      cout << "enter the value of radius of sphere";

      cin >> r\_025;

      volume(r\_025);

      break;

   case 2:

      cout << "enter the value of radius and height of cylinder ";

      cin >> r\_025 >> h\_025;

      volume(r\_025, h\_025);

      break;

   case 3:

      cout << "enter the value of 3 sides of triangle ";

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

      volume(a\_025, b\_025, c\_025);

   default:

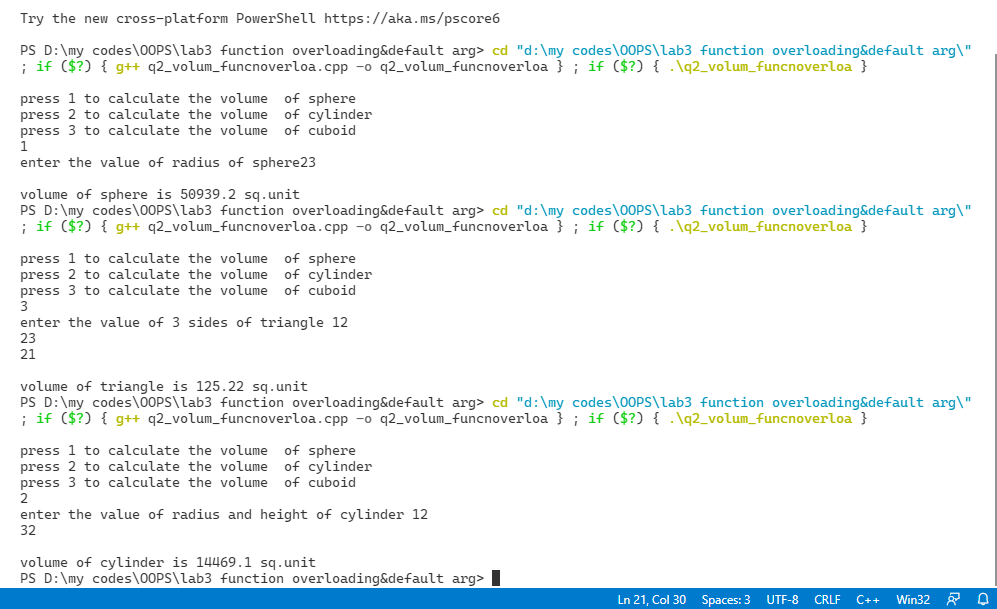
      break;

   }

   return 0;

}

**OUTPUT-Q2**



/\* Q3.WAP which displays a given character, n\_025 number of times, using a

function. When the n\_025 value is not provided, it should print the given

character 80 times. When both the character and n\_025 value is not provided, it

should print ‘\*’ character 80 times.

[Write the above program in two ways:-

-using function overloading.

-using default arguments.]

\*/

#include <iostream>

using namespace std;

/\*void charac(char c\_025 = '\*', int n\_025 = 80)

{

    for (int i = 0; i < n\_025; i++)

    {

        cout << i + 1 << "--->" << c\_025 << "\n";

    }

}\*/

void print(int t\_025)

{

    for (int i = 0; i < t\_025; i++)

    {

        cout << i + 1 << "--->" << '\*' << "\n";

    }

}

void print(char c\_025, int n\_025)

{

    for (int i = 0; i < n\_025; i++)

    {

        cout << i + 1 << "--->" << c\_025 << "\n";

    }

}

void print(char c\_025)

{

    for (int i = 0; i < 80; i++)

    {

        cout << i + 1 << "--->" << c\_025 << "\n";

    }

}

int main()

{

    int n\_025;

    cout << "what do you have ??\n"

         << "-----------------------------";

    cout << "\nif you have character then press 1"

         << "\n or if you have no. then press 2"

         << "\nand if you have nothing then press 3";

    cin >> n\_025;

    switch (n\_025)

    {

    case 1:

        char c\_025;

        cout << "enter the character";

        cin >> c\_025;

        print(c\_025);

        break;

    case 2:

        int t\_025;

        cout << "enter the number";

        cin >> t\_025;

        print(t\_025);

        break;

    case 3:

        print('\*', 80);

        break;

    default:

        break;

    }

    //charac();

    return 0;

}

OUTPUT-Q3



// Q4.WAP to find square and cube of a number using inline function.

//Inline int sq(int I) { return I\*I;}

#include <iostream>

using namespace std;

inline void sqr(int t\_025)

{

    cout << "sqr of the required no. is " << t\_025 \* t\_025;

}

inline void cube(int t\_025)

{

    cout << "cube of the required no. is " << t\_025 \* t\_025 \* t\_025;

}

int main()

{

    int t\_025;

    cout << "presss 1 to caluate sqr\n"

         << "press 2 to calculate cube\n";

    cin >> t\_025;

    switch (t\_025)

    {

    case 1:

    {

        int n025;

        cout << "enter the no.";

        cin >> n025;

        sqr(n025);

        break;

    }

    case 2:

    {

        int n025;

        cout << "enter the no.";

        cin >> n025;

        cube(n025);

        break;

    }

    default:

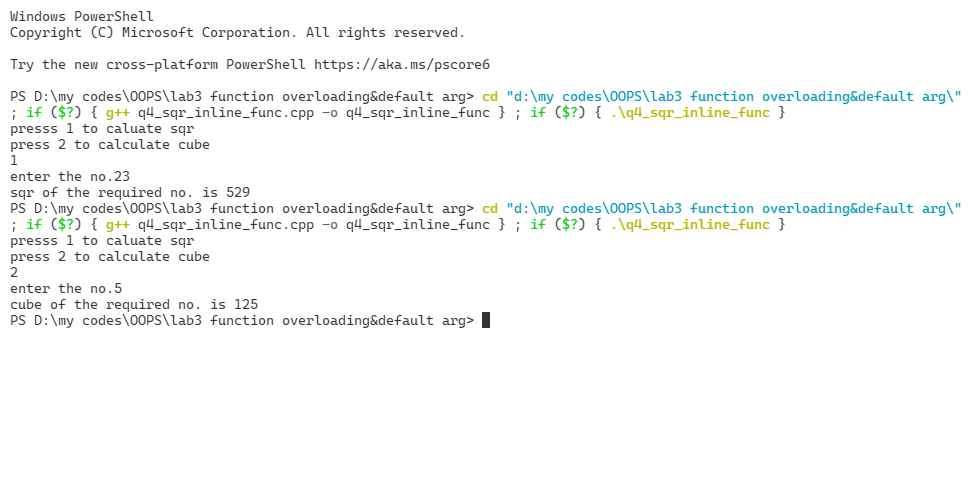
        break;

    }

    return 0;

}

OUTPUT-Q4



/\*Q5. WAP to increment the value of an argument given to function USING

INLINE function.

Inline int incr(int I) {

return ++i; }\*/

#include<iostream>

using namespace std;

inline void increm(int \*n025)

{

    cout<<"incremented value is "<< ++\*n025;

}

int main()

{int n025;

    cout<<"enter the no. you want to be incremented ";

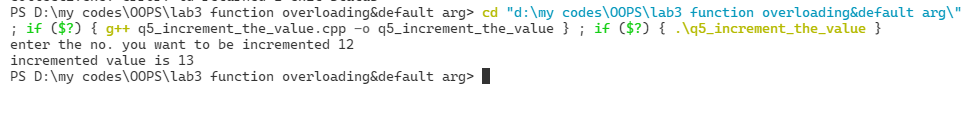
 cin>>n025;

 increm(&n025);

    return 0;

}

OUTPUT-Q5



/\*Q6.Write a\_025 program to create a\_025 class called COMPLEX and implement the following

overloading functions ADD that return a\_025 COMPLEX number.

a\_025) Complex ADD (int a\_025 , complex s2){

Complex temp;

Temp.real\_025=s2.real\_025+a\_025;

Temp.img\_025;

=s2.img\_025;

;

Return temp;

}

- where a\_025 is an integer (real\_025 part) and s2 is a\_025 complex number.

b) complex ADD (complex s1, complex s2) - where s1 and s2 are complex numbers.\*/

#include <iostream>

using namespace std;

class Complex

{

    int real\_025;

    int img\_025;

    ;

public:

    void input()

    {

        cin >> real\_025 >> img\_025;

        ;

    }

    Complex ADD(int a\_025, Complex s2)

    {

        Complex temp;

        temp.real\_025 = s2.real\_025 + a\_025;

        temp.img\_025 = s2.img\_025;

        ;

        return temp;

    }

    Complex ADD(Complex s1, Complex s2)

    {

        Complex temp;

        temp.real\_025 = s2.real\_025 + s1.real\_025;

        temp.img\_025 = s2.img\_025 + s1.img\_025;

        ;

        return temp;

    }

    void display()

    {

        cout << "\nThe added value is " << real\_025 << "+i" << img\_025;

        ;

    }

};

int main()

{

    int n\_025, a\_025;

    Complex s[2], c\_025;

    for (int i = 0; i < 2; i++)

    {

        cout << "enter" << i + 1 << " complex no.";

        s[i].input();

    }

    cout << "\npress 1 if you want to add one real no. to the 1st complex no."

         << "\npress 2 if you want to add one real no. to the 2nd complex no."

         << "\npress 3 if you want to add both the complex no.";

    cin >> n\_025;

    switch (n\_025)

    {

    case 1:

        cout << "enter a\_025 real  number";

        cin >> a\_025;

        c\_025 = c\_025.ADD(a\_025, s[0]);

        c\_025.display();

        break;

    case 2:

        cout << "enter a\_025 real number";

        cin >> a\_025;

        c\_025 = c\_025.ADD(a\_025, s[1]);

        c\_025.display();

        break;

    case 3:

        c\_025 = c\_025.ADD(s[0], s[1]);

        c\_025.display();

    default:

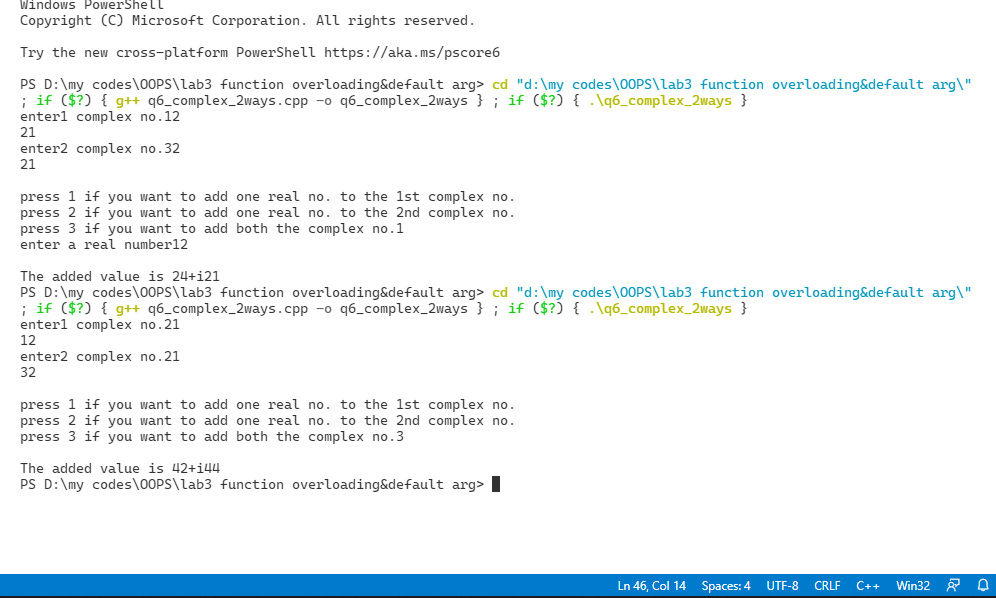
        break;

    }

    return 0;

}

OUTPUT-Q6



/\* Q7. Write a\_025 program to find the summation of three numbers by using one function

only with function name SUM having three arguments. If at run time one argument is

given to the function SUM, then second and third argument will be assumed by

default as 10 and 20 respectively. If two arguments are given at run time, then third

argument will be assumed as 20. Use function with default argument concepts.\*/

#include <iostream>

using namespace std;

void SUM(int a\_025, int b\_025 = 10, int c\_025 = 20)

{

    cout << "sum of 3 no. is " << a\_025 + b\_025 + c\_025;

}

int main()

{

    int n;

    int a\_025, b\_025, c\_025;

    cout << "press 1 if you have 1 no.s\n"

         << "press 2 if u have 2 no.s\n"

         << "press 3 if u have 3 no.s\n";

    cin >> n;

    switch (n)

    {

    case 1:

       int a\_025;

       cout << "enter 1 no.";

       cin >> a\_025;

       SUM(a\_025);

       break;

    case 2:

       cout << "enter 2 no.";

       cin >> a\_025 >> b\_025;

       SUM(a\_025, b\_025);

       break;

    case 3:

        cout << "enter 3 no.";

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

        SUM(a\_025, b\_025,c\_025);

        break;

    default:

        break;

    }

    return 0;

}

OUTPUT-Q7



/\*Q8.Write a\_025 program to demonstrate the concept of call-by-value, call-by-reference &amp;

call-by address by taking swapping of two numbers as an example.\*/

#include <iostream>

using namespace std;

void cbyvalue(int a\_025, int b\_025)

{

    int temp\_025 = a\_025;

    a\_025 = b\_025;

    b\_025 = temp\_025;

}

void cbyaddress(int &a\_025, int &b\_025)

{

    int temp\_025 = a\_025;

    a\_025 = b\_025;

    b\_025 = temp\_025;

}

void cbyrefrence(int \*a\_025, int \*b\_025)

{

    int \*temp\_025;

    \*temp\_025 = \*a\_025;

    \*a\_025 = \*b\_025;

    \*b\_025 = \*temp\_025;

}

int main()

{

    int a\_025, b\_025, n\_025;

    cout << "press 1 for call by value"

         << "\npress 2 for call by address"

         << "\npress 3 for call by refrence\n";

    cin >> n\_025;

    switch (n\_025)

    {

    case 1:

        cout << "enter 2 no.s\n";

        cin >> a\_025 >> b\_025;

        cbyvalue(a\_025, b\_025); //swaping doest haapen

        cout << "swap values are " << a\_025 << '&' << b\_025;

        break;

    case 2:

        cout << "enter 2 no.s\n";

        cin >> a\_025 >> b\_025;

        cbyaddress(a\_025, b\_025); //swap happens

        cout << "swap values are " << a\_025 << '&' << b\_025;

        break;

    case 3:

        cout << "enter 2 no.s\n";

        cin >> a\_025 >> b\_025;

        cbyrefrence(&a\_025, &b\_025); //swap happens

        cout << "swap values are " << a\_025 << '&' << b\_025;

        break;

    default:

        cout << "wrong input";

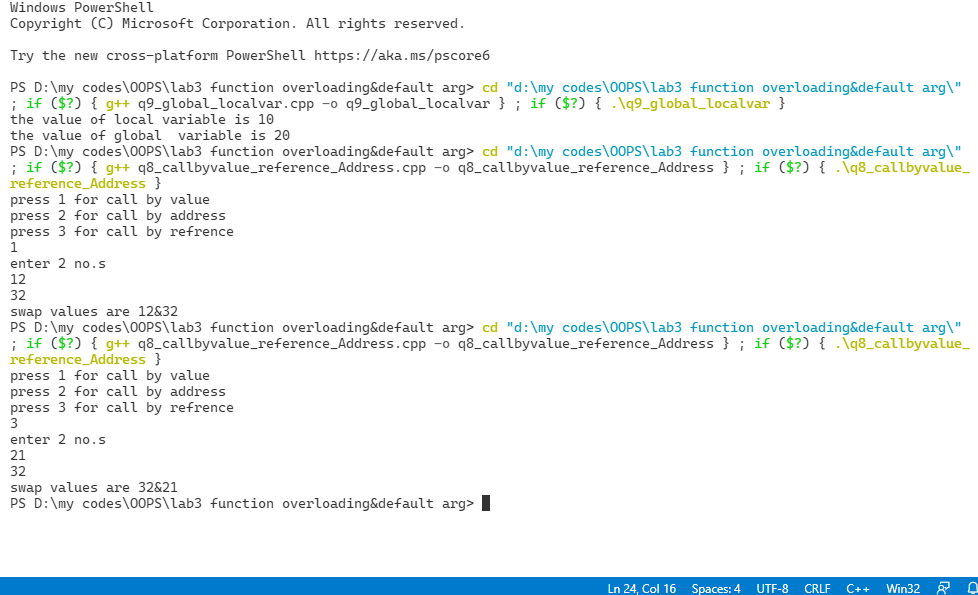
        break;

    }

    return 0;

}

OUTPUT-Q8



/\*Q9 Write a program to demonstrate the use of scope resolution operator(::) by

declaring same variable name globally and locally and display the value of global and

local variables.\*/

#include<iostream>

using namespace std;

int n\_025=20;//global variable

int main()

{

int n\_025=10;//local variable

 cout<<"the value of local variable is "<<n\_025;

 cout<<"\nthe value of global  variable is "<<::n\_025;

    return 0;

}

OUTPUT-Q9



**Name :Hitu Raj**

**Roll no. :2005025**

**Subject :OOP**

**Branch :CSE**

**3rd sem**