**INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN**



**Object Oriented Programming Practical File**

**BIT-204**

**Submitted by: Submitted to:**

NAME :- Meghavi Tomar Ms. VIBHA ROLL NO. :- 18601012020

CSE 2 B2

INDEX

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Experiment Name** | **Date** |
| 1 | Write a program to calculate the factorial of a number using functions. | 14-01-21 |
| 2 | Write a program to check if a number is prime or not using functions. | 21-01-21 |
| 3 | Write a program to calculate of a student using classes. It should maintain student’s record. Input is basic info of the student and make member functions as public to display  and read marks. | 28-01-21 |
| 4 | Write a program to show the use of i) Swapping using call by value, call by reference, call by address ii) Default value iii) and show function overloading. | 04-02-21 |
| 5 | Write a program to show the use of references, inline functions, macros, dynamic memory allocation using new and delete, scope and resolution operator. | 11-02-21 |
| 6 | Write a program to show addition of complex numbers using friend functions, objects as arguments by value and reference. | 18-02-21 |
| 7 | Write a program to perform operations on matrices using class like addition, subtraction, multiplication, transpose. | 25-02-21 |
| 8 | Write a program to calculate the area and perimeter of circle using classes. | 25-02-21 |
| 9 | Write a program to make a class for quadratic equation and find its roots. | 4-03-21 |
| 10 | WAP to add two timer (say t1,t2) using classes |  |
| 11 | WAP to implement constructor & destructor, constructor overloading, copy constructor by making an employee class. |  |
| 12 | WAP to show a comparison btw assignment operator & copy constructor. |  |

|  |  |  |
| --- | --- | --- |
| 13 | WAP to show different operator overloading. |  |
| 14 | WAP to show all kinds of inheritance with constructors and scope resolution. |  |
| 15 | WAP to show virtual function concept. |  |

# Experiment No. 1

**Aim**:

Write a program to calculate the factorial of a number using

functions.

# Code:

#include <iostream> #include <conio.h> using namespace std;

void factorial(int n)

{ int fact=1,i;

for(i=1; i<=n; i++)

{

fact=fact\*i;

}

cout<<"Factorial of "<<n<<" is: "<<fact;

}

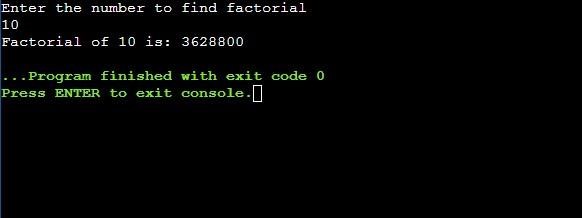
int main() { int n;

cout << "Enter the number to find factorial"

<< endl; cin>>n; factorial(n); getch(); return 0;

}

# Output:



**Experiment No. 2**

**Aim**:

Write a program to check if a number is prime or not using

functions.

# Code:

#include <iostream> using namespace std;

void isPrime(int n)

{

int i, flag = 0; for(i=2; i<=n/2; ++i)

{

if(n%i==0)

{

flag=1; break;

} } if (flag==0)

cout<<n<<" is a prime number"<<endl; else

cout<<n<<" is not a prime number"<<endl;

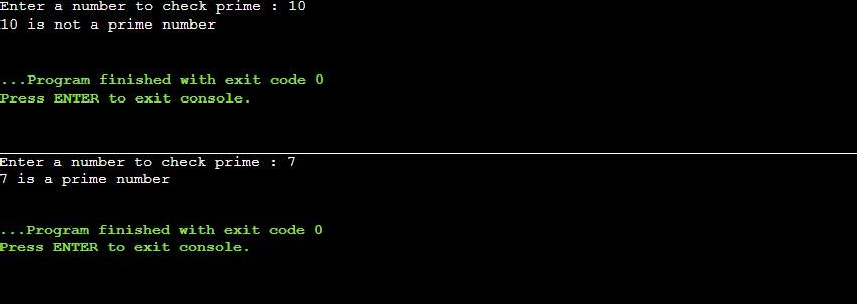
} int

main() { int n;

cout<<"Enter a number to check prime : "; cin>>n; isPrime(n); return 0;

}

# Output:



**Experiment No. 3**

**Aim**:

Write a program to calculate of a student using classes. It should maintain student’s record. Input is basic info of the student and make member functions as public to display and read marks.

# Code:

#include

<iostream> using namespace std; class marks

{ public: float f; int a,b,c,d,e,rollno;

char name[70];

void read()

{

cout << endl << "Enter the marks below:" << endl; cout << "Data Structures:

" ; cin >> a;

cout << "Discrete Structures: " ; cin >> b;

cout << "Database Management: " ; cin >> c;

cout << "Software engineering: " ; cin >> d;

cout << "Material Science: " ; cin >> e;

f =(a+b+c+d+e)/5;

}

void display()

{

cout<<"RollNo. :"<<rollno<<endl; cout<<"Name :"<<name<<endl;

cout<<"Marks obtained in different subjects:"<<endl; cout<<"Data structures

:"<<a<<endl; cout<<"Discrete structures

:"<<b<<endl; cout<<"Database Management

:"<<c<<endl; cout<<"Software Engineering:"<<d<<endl; cout<<"Material Science :"<<e<<endl; cout<<"Percentage obtained:"<< f<<"%"<<endl;

# Experiment No. 4

**Aim**:

}

};

int main ()

{

int i,n,k,l; marks pe[50];

cout<< "Enter the total no. of students in the class:"; cin>> n; if (n>50)

{ cout<<"Invalid no. of students";} else {

cout<<"---------------Enter Student Details "<<endl;

for(i=0;i<n;i++)

{

cout<<"Rollno.:"; cin>>pe[i].rollno; cout<<"Name :" ; cin>>pe[i].name; cout<<endl;

}

for (k=0;k<n;k++)

{

cout<<"Rollno. "<< pe[k].rollno; pe[k].read();

cout<<endl;

}

cout<<"-----------------Student Details-----------------

"<<endl; for (l=0;l<n;l++)

{

pe[l].display(); cout<<endl;

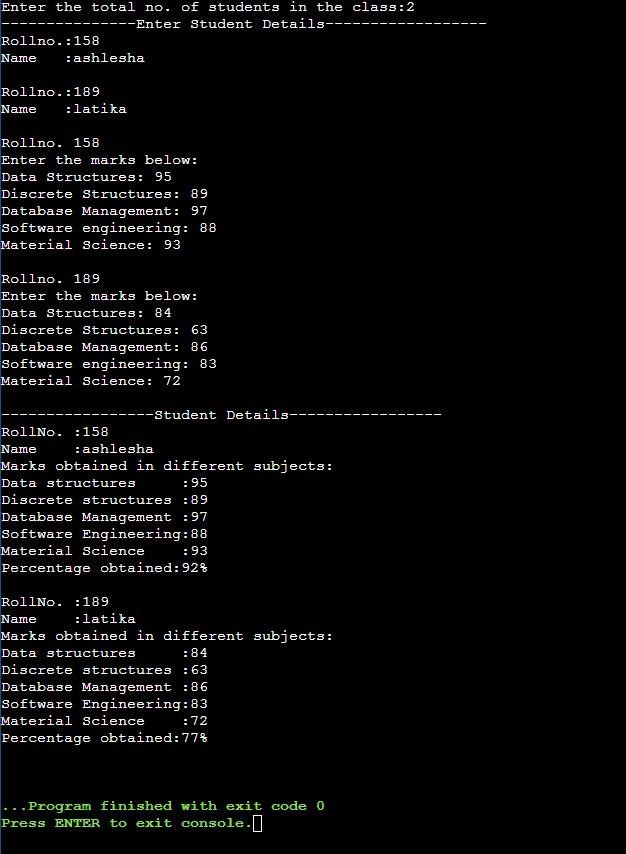
} }

return 0;

}

**Aim**:

# Output:



**Experiment No. 4**

Write a program to show the use of i) Swapping using call by value, call by reference, call by address ii) Default value iii) and show function overloading.

# Code:

#include <iostream> using namespace std;

int swap(int a, int b=0)

{

int c; c=a; a=b; b=c;

cout<<"The number a and b respectively are: "<<a<<" and "<<b<<endl; return 0;

}

int swap(float \*d, float \*e)

{ float f; f=\*d;

\*d=\*e;

\*e=f; return 0;

}

int add(int \*a, int \*b)

{

int c;

c=(\*a)+(\*b); return c;

}

int main()

{

int num1, num2; float num3, num4;

cout<<endl<<"----------SWAPPING THE NUMBERS-----------

"<<endl; cout<<endl<<"Enter the integers to be swapped below:"<<endl; cout<<"a: "; cin>>num1; cout<<"b: "; cin>>num2;

**Aim**:

cout<<endl<<"Enter the real numbers to be swapped

below:"<<endl; cout<<"d: "; cin>>num3; cout<<"e: ";

cin>>num4;

cout<<endl<<"----After Swapping "<<endl;

cout<<swap(&num3,&num4);

cout<<"The number d and e respectively are: "<<num3<<" and "<<num4<<endl; cout<<endl<<swap(num1,num2);

cout<<"(The numbers d and e respectively in the main function are: "<<num3<<" and

"<<num4<<")"<<endl;

cout<<"(The numbers a and b respectively in the main function are: "<<num1<<" and

"<<num2<<")"<<endl;

cout<<endl<<" ";

cout<<endl<< "Swapping after passing only one integer a in the function:"<<endl; cout<<swap(num1)<<endl;

cout<<"(The numbers a and b respectively in the main function are: "<<num1<<" and

"<<num2<<")"<<endl;

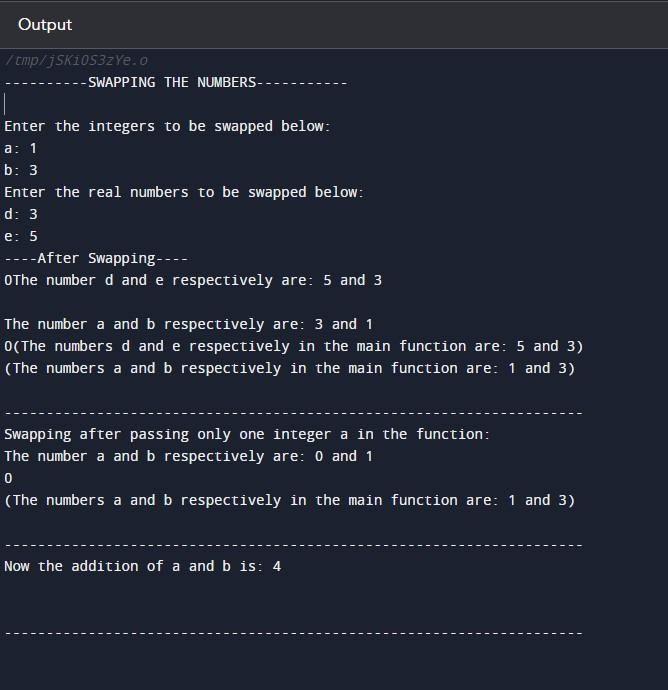
cout<<endl<<" ";

cout<<endl<<"Now the addition of a and b is: "<<add(&num1,&num2)<<endl<<endl; cout<<endl<<"--------------------------

-------------------------------------------"; return 0;

}

# Output:



**Experiment No.**

**Aim**:

# 5

Write a program to show the use of references, inline functions, macros, dynamic memory allocation using new and delete, scope and resolution operator.

# Code:

#include <iostream> using namespace std;

int s=40;

#define Print\_square cout<<endl<<"Square of local variable s is: "<<s\*s; inline int sum (int a, int b)

{

return a+b;

}

void increment (int& s)

{ s++;

}

int main ()

{ int a,b,c;

cout<<endl<<"Enter the numbers to be added below: "<<endl; cout<<"Enter a: "; cin>>a; cout<<"Enter b: "; cin>>b; c=sum(a,b); cout<<endl<<"The sum of a and b is: "<<c;

int s; s=12;

cout<<endl<<"

"<<endl;

Print\_square

cout<<endl<<"s as local variable is: "<<s; increment(s);

cout<<endl<<endl<<"-------After incrementing the value of s ";

Print\_square

cout<<endl<<endl<<"s as local variable is: "<<s; cout<<endl<<"s as global variable is: "<<::s;

cout<<endl<<"

"<<endl; int \*d; d = new int; \*d=10; cout<<endl<<"Pointer d points to a value of: "<<\*d<<endl; delete d;

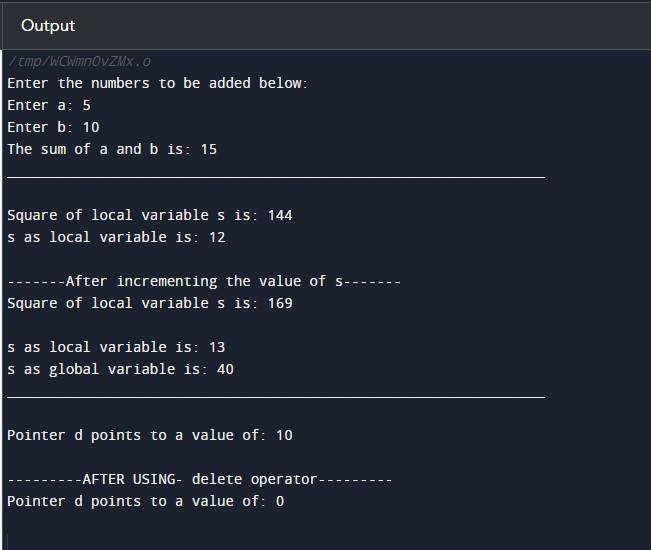
cout<<endl<<"---------AFTER USING- delete operator ";

cout<<endl<<"Pointer d points to a value of: "<<\*d<<endl<<endl;

return 0;

}

# Output:



**Experiment No.**

**Aim**:

# 6

Write a program to show addition of complex numbers using friend functions, objects as arguments by value and reference.

# Code:

#include <iostream> using namespace std;

class cl

{

private:

int sum1, sum2; float sumi, sumr; public:

int img1, img2, real1, real2;

float img\_1, img\_2, real\_1, real\_2;

friend void complex1(cl d); friend int complex2(cl &d);

};

void complex1 (cl d)

{

d.sum1 = d.img1 + d.img2;

d.sum2 = d.real1 + d.real2; cout<<endl<<endl<<"Sum of the Two complex numbers in the function is:

"<<d.sum2<<" + i"<<d.sum1;

}

int complex2 (cl &d)

{

d.sumi = d.img\_1 + d.img\_2;

d.sumr = d.real\_1 + d.real\_2; cout<<endl<<endl<<"Sum of the Two complex numbers in the function is:

"<<d.sumr<<" + i"<<d.sumi;

}

int main()

{

cl d;

cout<<endl<<"Enter the complex numbers (int values) below:"<<endl; cout<<endl<<"Complex Number 1:"<<endl;

cout<<"Real part: "; cin>>d.real1; cout<<"Imaginary part: "; cin>>d.img1;

cout<<endl<<"Complex Number 2:"<<endl; cout<<"Real part: ";

cin>>d.real2;

cout<<"Imaginary part: "; cin>>d.img2;

cout<<endl<<"The two complex numbers are: "<<d.real1<<" + i"<<d.img1<<" and

"<<d.real2<<" + i"<<d.img2; complex1(d);

cout<<endl<<"

";

cout<<endl<<endl<<"Enter the complex numbers (float values) below:"<<endl; cout<<endl<<"Complex Number 1:"<<endl;

cout<<"Real part: "; cin>>d.real\_1; cout<<"Imaginary part: "; cin>>d.img\_1;

cout<<endl<<"Complex Number 2:"<<endl; cout<<"Real part: ";

cin>>d.real\_2; cout<<"Imaginary part: "; cin>>d.img\_2;

cout<<endl<<"The two complex numbers are: "<<d.real\_1<<" + i"<<d.img\_1<<" and

"<<d.real\_2<<" + i"<<d.img\_2; complex2(d);

cout<<endl<<"

\_"

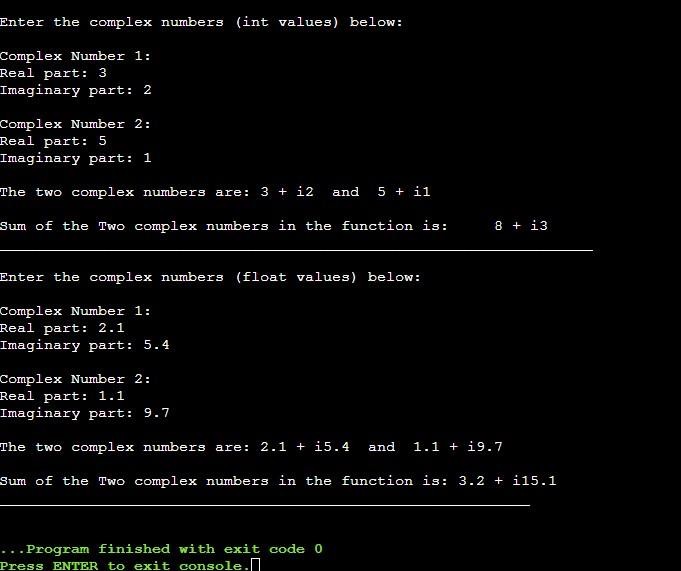
<<endl; return 0;

}

# Output:

**Experiment No.**

**Aim**:



# Experiment No.

**7**

**Aim**: Write a program to perform operations on matrices using class like addition, subtraction, multiplication, transpose.

# Code:

#include <iostream> using namespace std;

class arr { public:

int a[20][20],

b[20][20];

};

void trans(int a[][20], int m, int n)

{

int b[20][20]; for (int i=0; i<m; i++)

{

for (int j=0; j<n; j++)

{

b[j][i]=a[i][j];

}

}

cout<<endl<<"The transpose: A'"<<endl; for(int i=0; i<n; i++)

{

for (int j=0; j<m; j++)

{

cout<<b[i][j]<<" ";

}

cout<<endl;

} }

void mul(int a[][20], int b[][20], int m, int n, int q)

{ int c[20][20];

cout<<endl<<"The product: A\*B"<<endl; for (int i=0; i<m; i++)

{

for (int j=0; j<q; j++)

{

c[i][j] =0;

{

for (int k=0; k<n; k++)

{

c[i][j]=c[i][j]+(a[i][k]\*b[k][j]);

}

}

cout<<c[i][j]<<" ";

}

cout<<endl;

}

}

void add( int a[][20], int b[][20], int m, int n)

{ int c[20][20];

for (int i=0; i<m; i++)

{

for (int j=0; j<n; j++)

{

c[i][j]=b[i][j]+a[i][j];

}

}

cout<<endl<<"The sum: A+B"<<endl; for(int i=0; i<m; i++)

{

for (int j=0; j<n; j++)

{

cout<<c[i][j]<<" ";

}

cout<<endl;

}

}

void sub(int a[][20], int b[][20], int m, int n)

{ int c[20][20];

for (int i=0; i<m; i++)

{

for (int j=0; j<n; j++)

{

c[i][j]=a[i][j]-b[i][j];

}

}

cout<<endl<<"The difference: A-B"<<endl; for(int i=0; i<m; i++)

{

for (int j=0; j<n; j++)

{

cout<<c[i][j]<<" ";

}

cout<<endl;

}

}

int main()

{ int

m,n,p,q; arr dr;

cout<<endl<<"Enter the number of rows in matrix A: "; cin>>m;

cout<<"Enter the number of columns in matrix A: "; cin>>n;

cout<<endl<<"Enter the number of rows in matrix B: "; cin>>p;

cout<<"Enter the number of columns in matrix B: "; cin>>q;

cout<<endl<<"Enter the elements of matrix A below:"<<endl; for(int i=0; i<m; i++)

{

for (int j=0; j<n; j++)

{

cout<<"A["<<i+1<<"]["<<j+1<<"]: ";

cin>>dr.a[i][j];

}

}

cout<<endl<<"Enter the elements of matrix B below:"<<endl; for(int i=0; i<p; i++)

{

for (int j=0; j<q; j++)

{

cout<<"B["<<i+1<<"]["<<j+1<<"]: ";

cin>>dr.b[i][j];

}

}

cout<<endl<<" Matrices A and

B "<<endl; cout<<endl<<" A: "<<endl; for(int i=0; i<m; i++)

{

for (int j=0; j<n; j++)

{

cout<<dr.a[i][j]<<" ";

}

cout<<endl;

}

cout<<endl<<" B: "<<endl;

for(int i=0; i<p; i++)

{

for (int j=0; j<q; j++)

{

cout<<dr.b[i][j]<<" ";

}

cout<<endl;

}

cout<<endl<<" Addition of matrices A + B "<<endl;

if (m==p&&n==q)

{

add(dr.a,dr.b,m,n);

}

else

{

cout<<endl<<"Addition of matrices is not possible!!!!";

}

cout<<endl<<endl<<" Subtraction of matrices A - B "<<endl;

if (m==p&&n==q)

{

sub(dr.a,dr.b,m,n);

}

else

{

cout<<endl<<"Subtraction of matrices is not possible!!!!";

}

cout<<endl<<endl<<" Multiplication of matrices A\*B "<<endl;

if (n==p)

{

mul(dr.a,dr.b,m,n,q);

}

else

{

cout<<endl<<"Multiplication of matrices is not possible!!!!";

}

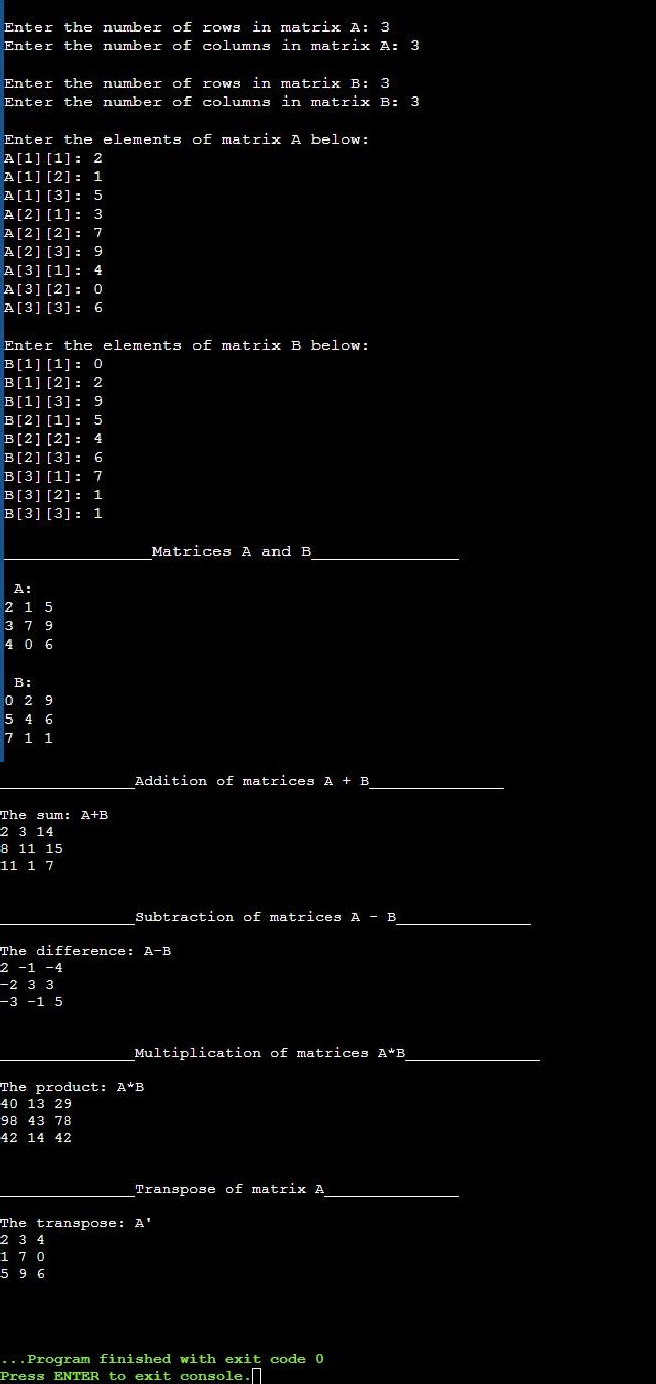
cout<<endl<<endl<<" Transpose of matrix A "<<endl;

trans(dr.a,m,n);

cout<<endl<<endl; return 0;

}

# Output:



**Experiment No. 8**

**Aim**: Write a program to calculate the area and perimeter of circle using classes.

# Code:

#include<iostrea m> #include<conio.h

> using

namespace std; class Circle

{

private: float r; public:

void input()

{

cout<<"Enter radius of circle: "; cin>>r;

}

void Area()

{

float a=22.0/7.0\*r\*r; cout<<"\nArea of circle ="<<a;

}

void Circumference()

{

float p=2\*22.0/7.0\*r; cout<<"\nCircumference of circle ="<<p;

} }; int main() { Circle c1;

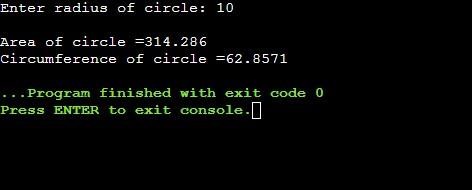
c1.input();

c1.Area(); c1.Circumference ();

return 0;

}

# Output:



**Experiment No. 9**

**Aim**: Write a program to make a class for quadratic equation and find its roots.

# Code:

#include

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

class roots { int a, b, c; float r1, r2; public:

void getdata()

{

cout << "Enter value of coefficient of x^2: "; cin >> a;

cout << "Enter value of coefficient of x: "; cin >> b;

cout << "Enter value of coefficient of 1: "; cin >> c;

}

int determinant()

{

int d = b \* b; d -= (4 \* a \* c); return d;

}

void checkdeterminant(int d)

{

if (d == 0)

{

cout << "Real and equal

roots\n"; r1 = (-1 \* b); r1 /= (2 \* a);

r2 = r1;

cout << "Roots : " << r1 << " and " << r2 << endl;

}

else if (d > 0)

{

cout << "Real and distinct

roots\n"; r1 = (-1 \* b) + sqrt(d); r1 /= (2 \* a);

r2 = (-1 \* b) - sqrt(d); r2 /= (2 \* a);

cout << "Roots : " << r1 << " and " << r2 << endl;

}

else

{

cout << "Imaginary roots" << endl << endl; r1 = (-b)/(2\*a); r2 = (sqrt(-d))/(2\*a);

cout << "Roots : " << r1 << " + i" << r2 ; cout << " and " << r1 << " - i" << r2 << endl<<endl;

}

}

};

int main()

{

roots r;

r.getdata(); cout << endl; int d = r.determinant(); r.checkdeterminant(d

);

cout << endl;

return 0;

}

# Output:

