

**AIM:** Write a C++ program to find the roots of a quadratic equation.

**PROGRAM:**

```
#include<iostream>
#include<cmath>
using namespace std;
int main()
{
    double a,b,c,d,r1,r2; // Variable declarations
    cout<<"enter a,b,c";
    cin>>a>>b>>c; // Input coefficients
    d=(b*b)-(4*a*c); // Calculate discriminant
    // Check nature of roots based on discriminant
    if(d==0)
    {
        // Case 1: Discriminant = 0 → roots are real and equal
        cout<<"roots are real and equal"<<endl;
        r1=-b/(2*a);
        r2=-b/(2*a);
        cout<<"r1="<<r1<<endl<<"r2"<<r2;
    }
    else if(d>0)
    {
        // Case 2: Discriminant > 0 → roots are real and distinct
        cout<<"roots are real and distinct"<<endl;
        r1=(-b+sqrt(d))/(2*a);
        r2=(-b-sqrt(d))/(2*a);
        cout<<"r1="<<r1<<endl<<"r2="<<r2;
    }
    else
    {
        // Case 3: Discriminant < 0 → roots are imaginary (no real solution)
        cout<<"roots are imaginary";
```

```
}
```

```
return 0;
```

```
}
```

**OUTPUT:**

The screenshot shows a terminal window titled 'C:\Users\krish\Desktop\roots'. The window contains the following text output:

```
enter a,b,c
2 4 2
roots are real and equal
r1=-1
r2=1
-----
Process exited after 9.988 seconds with return value 0
Press any key to continue . . . |
```

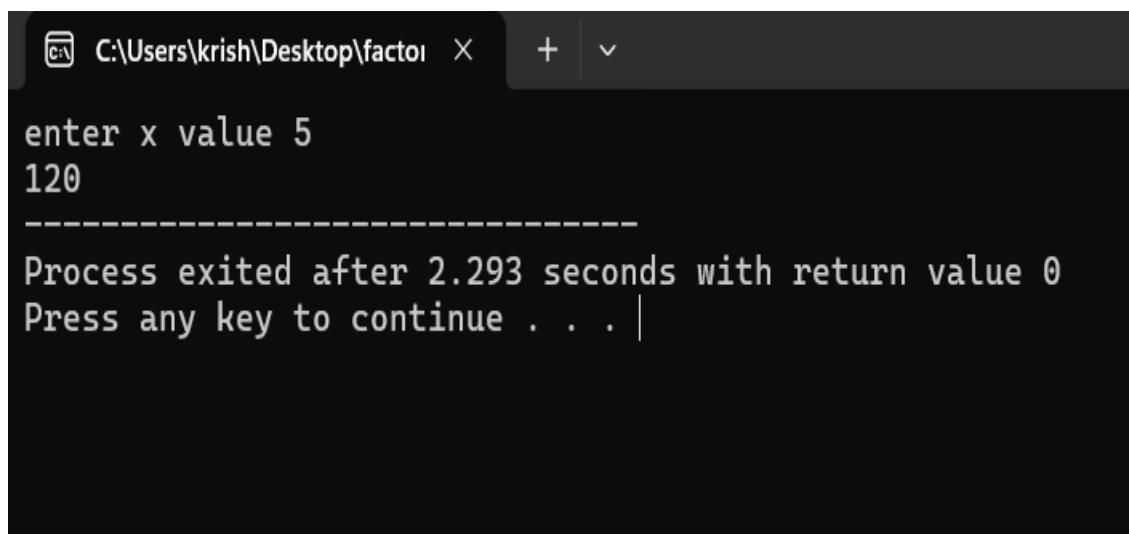
**AIM:** Write a C++ program to find factorial of a given number using recursion.

**PROGRAM:**

```
#include<iostream>
using namespace std;
int factorial(int n) // Function to calculate factorial using recursion
{
    // Base case: factorial of 1 is 1
    if(n==1)
        return 1;
    else
        return(n*factorial(n-1)); // Recursive case: n! = n * (n-1)!

}
int main()
{
    int x; // Variable to store user input number
    cout<<"enter x value";
    cin>>x;
    cout<<factorial(x); //printing value by calling fuction
    return 0;
}
```

**OUTPUT:**



The screenshot shows a terminal window with the following text output:

```
C:\Users\krish\Desktop\factorial + 
enter x value 5
120
-----
Process exited after 2.293 seconds with return value 0
Press any key to continue . . . |
```

**AIM:** Write a C++ program to implement scope resolution and namespaces.

**PROGRAM:**

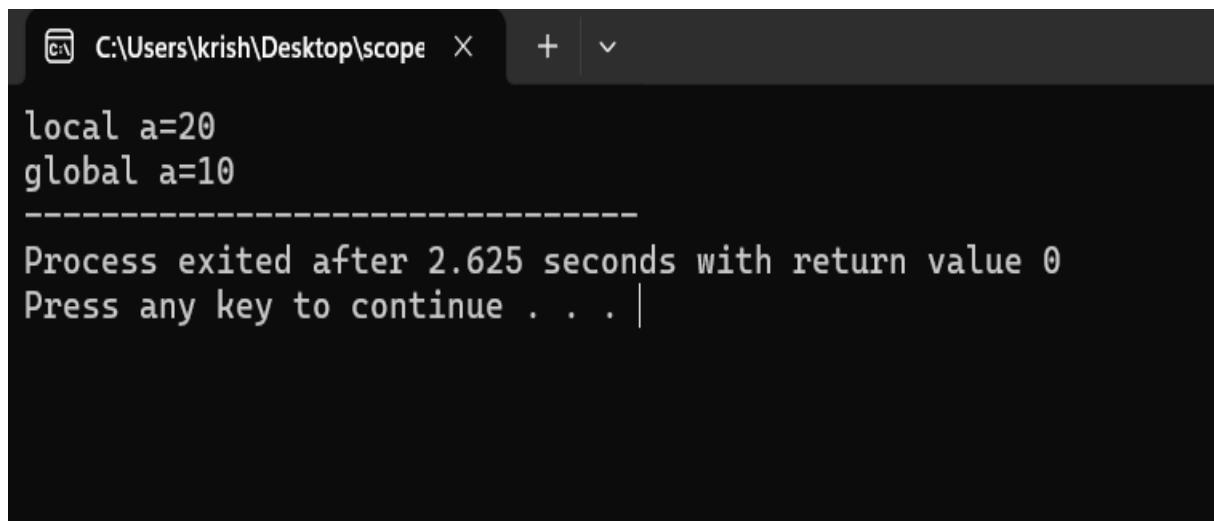
```
#include<iostream>

using namespace std;

int a=10;//global variable

int main()
{
    int a=20;//local variable
    cout<<"local a ="<<a<<endl;//printing local variable
    cout<<"global a ="<<::a;//printing global variable
    return 0;
}
```

**OUTPUT:**



```
C:\Users\krish\Desktop\scope + ▾
local a=20
global a=10
-----
Process exited after 2.625 seconds with return value 0
Press any key to continue . . . |
```

**AIM:** Write a C++ program to implement scope resolution and namespaces.

**PROGRAM:**

```
#include<iostream>

using namespace std;

// first name space

namespace one

{

    void display()

    {

        int a=5;

        cout<<a;

    }

}

// second name space

namespace two

{

    void display()

    {

        string m="cse-";

        cout<<m;

    }

}

int main()

{

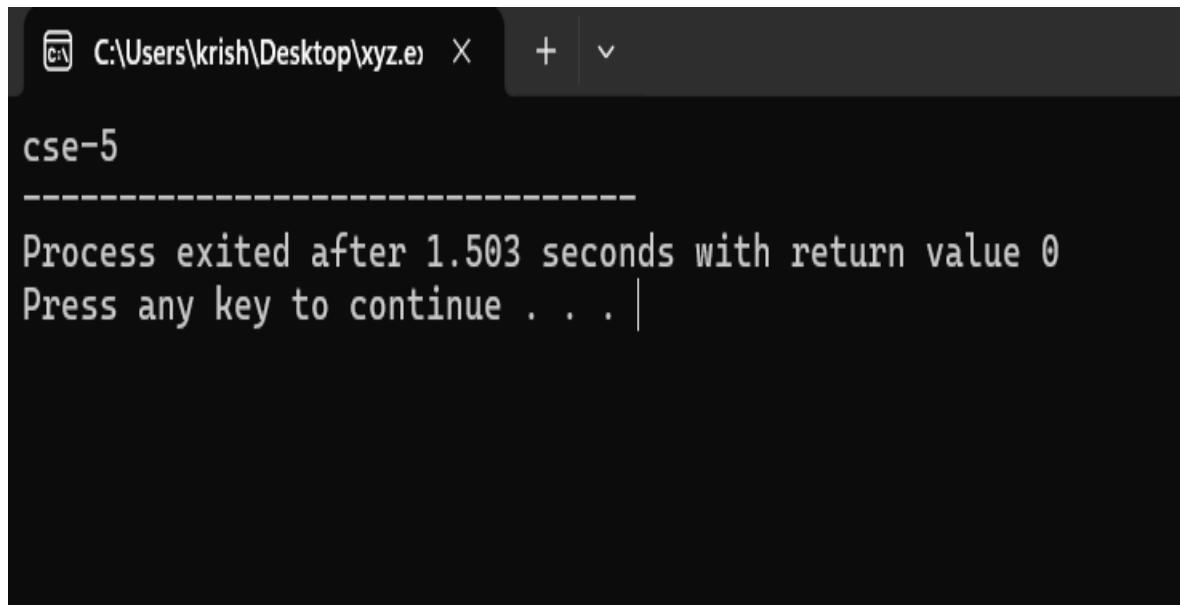
    two::display();// accessing variable from second name space.

    one::display();// accessing variable from first name space.

    return 0;

}
```

**OUTPUT:**



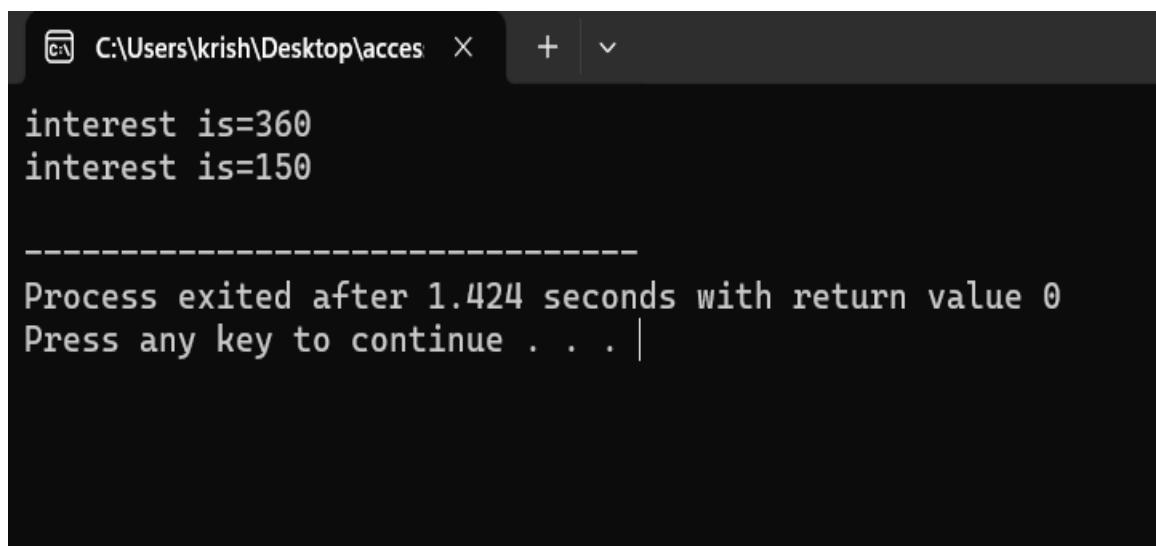
A screenshot of a terminal window titled 'C:\Users\krish\Desktop\xyz.e'. The window contains the following text:  
cse-5  
-----  
Process exited after 1.503 seconds with return value 0  
Press any key to continue . . . |

**AIM:** Write a C++ program to illustrate the use of default arguments and access specifiers

**PROGRAM:**

```
#include<iostream>
using namespace std;
// Function to calculate Simple Interest
// Default arguments: p=1000, t=5, r=3
// Formula: (p * t * r) / 100
void SI(int p=1000,int t=5,float r=3)
{
    cout<<"interest is="<<(p*t*r)/100<<endl;
}
int main()
{
    SI(3000,4);//function call with only 2 arguments
    SI();//function call with no arguments
    return 0;
}
```

## OUTPUT:



A screenshot of a terminal window titled "C:\Users\krish\Desktop\acces". The window contains the following text:

```
interest is=360
interest is=150
-----
Process exited after 1.424 seconds with return value 0
Press any key to continue . . . |
```

**AIM:** Write a C++ program to illustrate the use of default arguments and access specifiers.

**PROGRAM:**

```
#include <iostream>

using namespace std;

// Define a class named Sample

class Sample {

private:

    int rollNo; // Private data member

public:

    string name; // Public data member

    // Parameterized constructor

    // Used to initialize 'name' and 'rollNo' when object is created

    Sample(string n, int r) {

        name = n; // Assign parameter n to member 'name'

        rollNo = r; // Assign parameter r to member 'rollNo'

    }

    // Member function to display roll number

    void show() {

        cout << "Roll No (Private using member function): " << rollNo << endl;

    }

};

int main() {

    // Create object 's' of class Sample and initialize it using constructor

    Sample s("Manohar", 250);

    // Accessing public data member directly

    cout << "Student Name (public): " << s.name << endl;

    s.show();

    return 0;
}
```

}

**OUTPUT:**

```
C:\Users\krish\Desktop\cpp1.c  X  +  ▾

Student Name (public): Manohar
Roll No (Private using member function): 250

-----
Process exited after 7.061 seconds with return value 0
Press any key to continue . . .
```