# Programming Laboratory-I

## Assignment No-5

### (Virtual functions and namespace)

**2020BTECS00005**

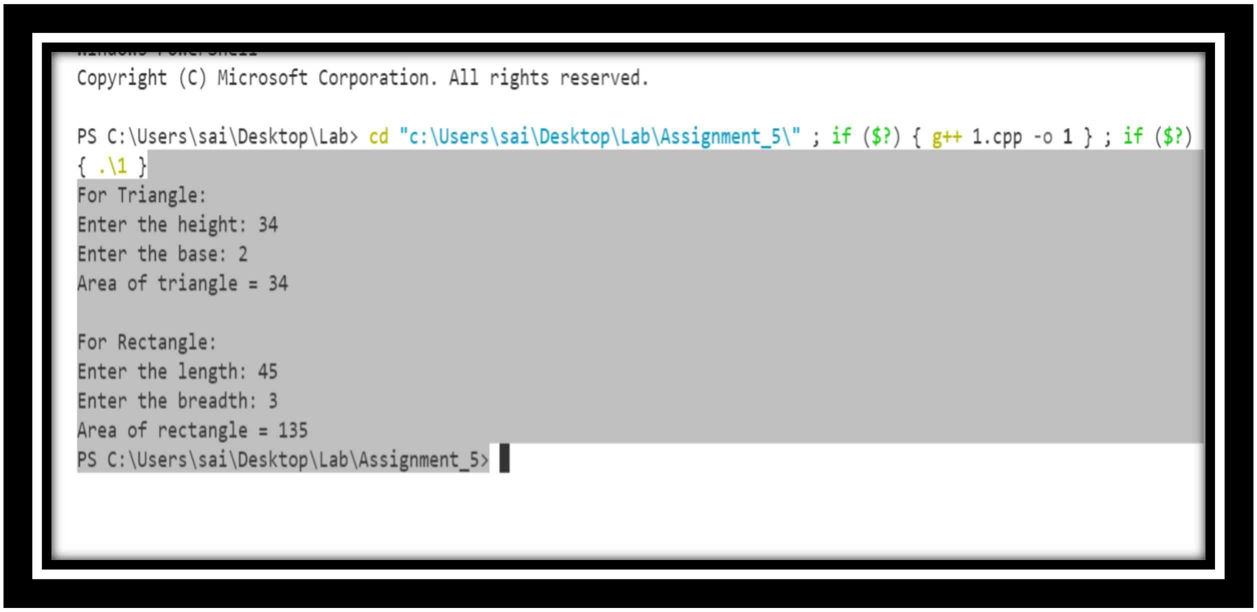
**Sanket Shivaji Jadhav**

1. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data() to initialize base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit their requirements.
   Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively, and display the area. Remember the two values given as input will be treated as lengths of two sides in the case of rectangles, and as base and height in the case of triangles, and used as follows:
   Area of rectangle = x * y
   Area of triangle = 1/2 * x * y

```
#include <bits/stdc++.h>

using namespace std;
class shape
{
public:
    double h, b;
    void get_data(double x, double y)
    {
        h = x;
        b = y;
    }
    virtual void display_area() {}
};
class Triangle : public shape
{
```

```
public:
    void display_area()
    {
        cout << "Area of triangle = " << (h * b) / 2 <<
endl;
    }
};
class Rectangle : public shape
{
public:
    void display_area()
    {
        cout << "Area of rectangle = " << h * b << endl;
    }
};
int main()
{
    shape *s;
    Triangle t;
    Rectangle r;
    double x, y;
    cout << "For Triangle:\n";
    cout << "Enter the height: ";
    cin >> x;
    cout << "Enter the base: ";
    cin >> y;
    t.get_data(x, y);
    s = &t;
    s->display_area();
    cout << "\nFor Rectangle:\n";
    cout << "Enter the length: ";
    cin >> x;
    cout << "Enter the breadth: ";
    cin >> y;
    r.get_data(x, y);
    s = &r;
    s->display_area();
        return 0;
}
```

Output:



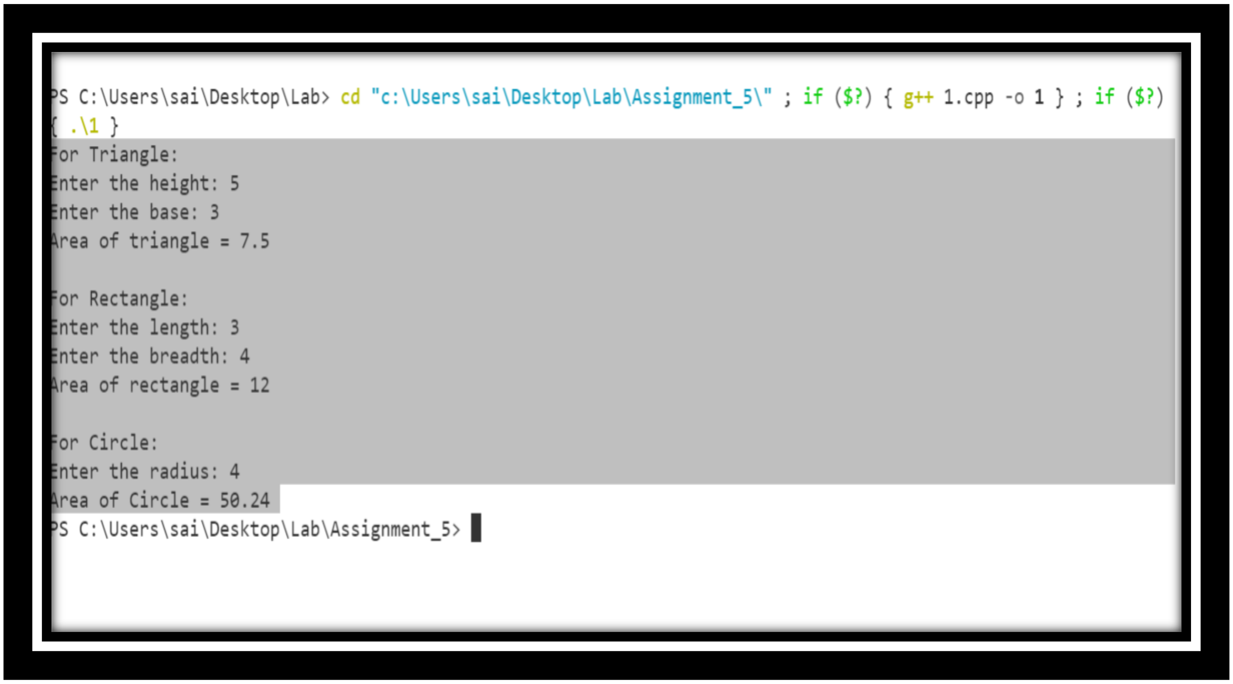
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\sai\Desktop\Lab> cd "c:\Users\sai\Desktop\Lab\Assignment_5\" ; if ($?) { g++ 1.cpp -o 1 } ; if ($?)
{ .\1 }
For Triangle:
Enter the height: 34
Enter the base: 2
Area of triangle = 34

For Rectangle:
Enter the length: 45
Enter the breadth: 3
Area of rectangle = 135
PS C:\Users\sai\Desktop\Lab\Assignment_5>
```

2. Extend the above program to display the area of circles. This requires addition of a new derived class 'circle' that computes the area of a circle. Remember, for a circle we need only one value, its radius, but the get_data() function in the base class requires two values to be passed. (Hint: Make the second argument of get_data() function as a default one with zero value.)

```
 #include <bits/stdc++.h>

using namespace std;
class shape
{
public:
    double h, b;
    void get_data(double x, double y)
    {
        h = x;
        b = y;
    }
    virtual void display_area() {}
};
class Triangle : public shape
{
```

```
public:
    void display_area()
    {
        cout << "Area of triangle = " << (h * b) / 2 <<
endl;
    }
};
class Rectangle : public shape
{
public:
    void display_area()
    {
        cout << "Area of rectangle = " << h * b << endl;
    }
};
class Circle : public shape
{
public:
    void display_area()
    {
        cout << "Area of Circle = " << 3.14 * h * h << endl;
    }
};
int main()
{
    shape *s;
    Triangle t;
    Rectangle r;
    Circle c;
    double x, y;
    cout << "For Triangle:\n";
    cout << "Enter the height: ";
    cin >> x;
    cout << "Enter the base: ";
    cin >> y;
    t.get_data(x, y);
    s = &t;
    s->display_area();
    cout << "\nFor Rectangle:\n";
    cout << "Enter the length: ";
    cin >> x;
    cout << "Enter the breadth: ";
    cin >> y;
    r.get_data(x, y);
    s = &r;
    s->display_area();
    cout << "\nFor Circle:\n";
    cout << "Enter the radius: ";
    cin >> x;
    c.get_data(x, 0);
    s = &c;
    s->display_area();

    return 0;
}
```

Output:



```
PS C:\Users\sai\Desktop\Lab> cd "c:\Users\sai\Desktop\Lab\Assignment_5\" ; if ($?) { g++ 1.cpp -o 1 } ; if ($?)
{ .\1 }
For Triangle:
Enter the height: 5
Enter the base: 3
Area of triangle = 7.5

For Rectangle:
Enter the length: 3
Enter the breadth: 4
Area of rectangle = 12

For Circle:
Enter the radius: 4
Area of Circle = 50.24
PS C:\Users\sai\Desktop\Lab\Assignment_5>
```

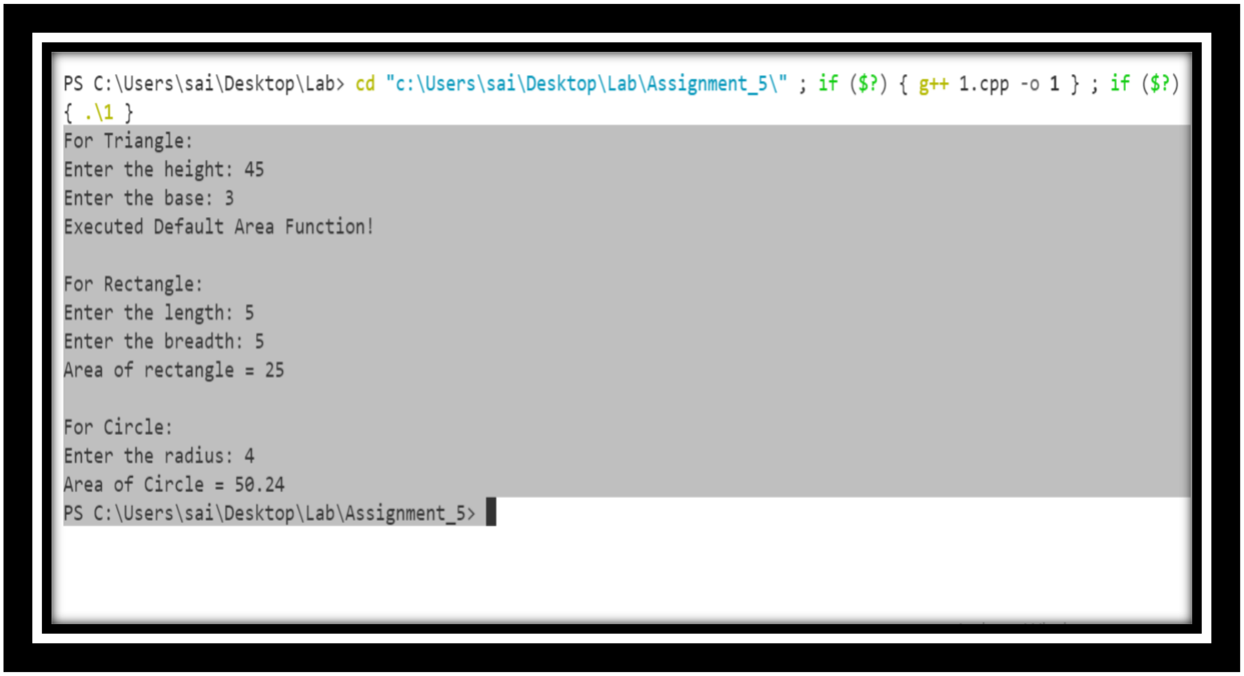
3. Run the above program with the following modifications:

(a) Remove the definition of display area() from one of the derived classes.
(b) In addition to the above change, declare the display_area() as virtual in the base class shape.

```
#include <iostream>

using namespace std;
class shape
{
public:
    double h, b;
    void get_data(double x, double y)
    {
        h = x;
        b = y;
    }
    virtual void display_area()
    {
        cout << "Executed Default Area Function!\n";
    }
};
class Triangle : public shape
{
public:
};
class Rectangle : public shape
{
public:
    void display_area()
    {
        cout << "Area of rectangle = " << h * b << endl;
    }
};
class Circle : public shape
{
public:
    void display_area()
    {
        cout << "Area of Circle = " << 3.14 * h * h << endl;
    }
```

```cpp
};
int main()
{
    shape *s;
    Triangle t;
    Rectangle r;
    Circle c;
    double x, y;
    cout << "For Triangle:\n";
    cout << "Enter the height: ";
    cin >> x;
    cout << "Enter the base: ";
    cin >> y;
    t.get_data(x, y);
    s = &t;
    s->display_area();


    cout << "\nFor Rectangle:\n";
    cout << "Enter the length: ";
    cin >> x;
    cout << "Enter the breadth: ";
    cin >> y;
    r.get_data(x, y);
    s = &r;
    s->display_area();


    cout << "\nFor Circle:\n";
    cout << "Enter the radius: ";
    cin >> x;
    c.get_data(x, 0);
    s = &c;
    s->display_area();

    return 0;
}
```

Output:



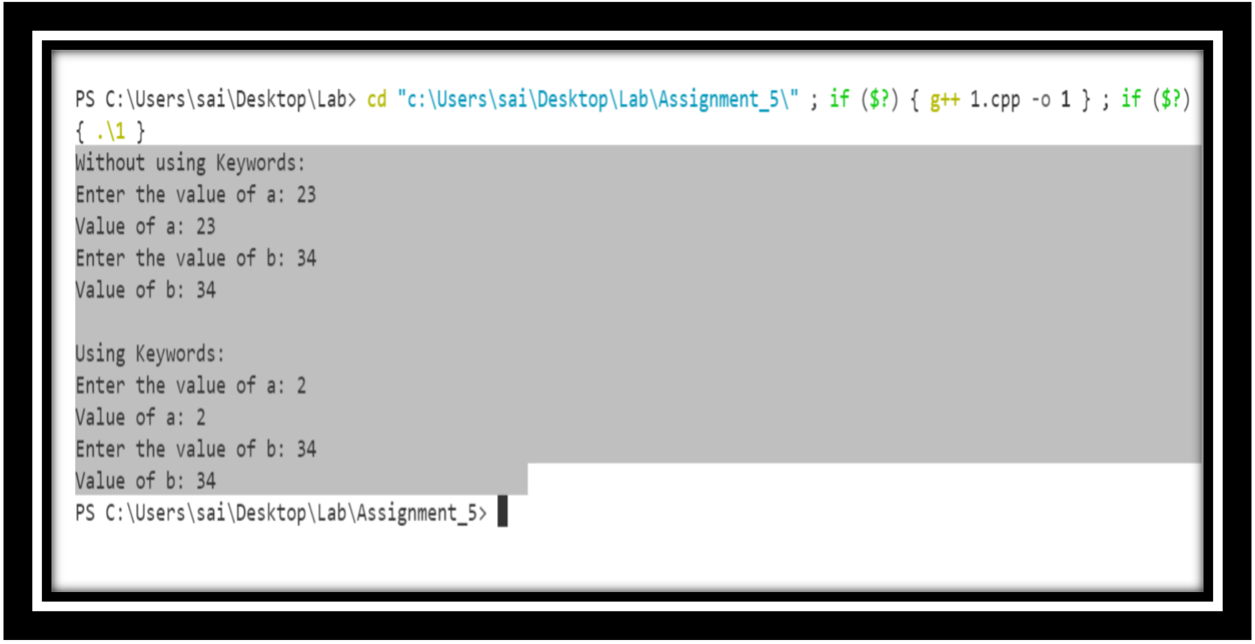
```
PS C:\Users\sai\Desktop\Lab> cd "c:\Users\sai\Desktop\Lab\Assignment_5\" ; if ($?) { g++ 1.cpp -o 1 } ; if ($?)
{ .\1 }
For Triangle:
Enter the height: 45
Enter the base: 3
Executed Default Area Function!

For Rectangle:
Enter the length: 5
Enter the breadth: 5
Area of rectangle = 25

For Circle:
Enter the radius: 4
Area of Circle = 50.24
PS C:\Users\sai\Desktop\Lab\Assignment_5> 
```

4. Create two namespaces as NMS1 and NMS2, define variable ‘a’ and function seta() and geta () in NMS1 and define variable ‘b’ and function setb() and getb () in NMS2. Write a program to
   a. Assign and display data **without using keyword.**
   b. Assign and display data **with using keyword.**

```
 #include <iostream>
using namespace std;
namespace NMS1
{
    int a;
    void seta()
    {
        cout << "Enter the value of a: ";
        cin >> a;
    }
    void geta()
    {
        cout << "Value of a: " << a << endl;
    }
}
```

```cpp
namespace NMS2
{
    int b;
    void setb()
    {
        cout << "Enter the value of b: ";
        cin >> b;
    }
    void getb()
    {
        cout << "Value of b: " << b << endl;
    }

}
int main()
{
    cout << "Without using Keywords:\n";

    NMS1 ::seta();
    NMS1 ::geta();
    NMS2 ::setb();
    NMS2 ::getb();

    cout << "\nUsing Keywords:\n";
    using namespace NMS1;
    seta();
    geta();

    using namespace NMS2;
    setb();
    getb();

    return 0;
}
```

Output:



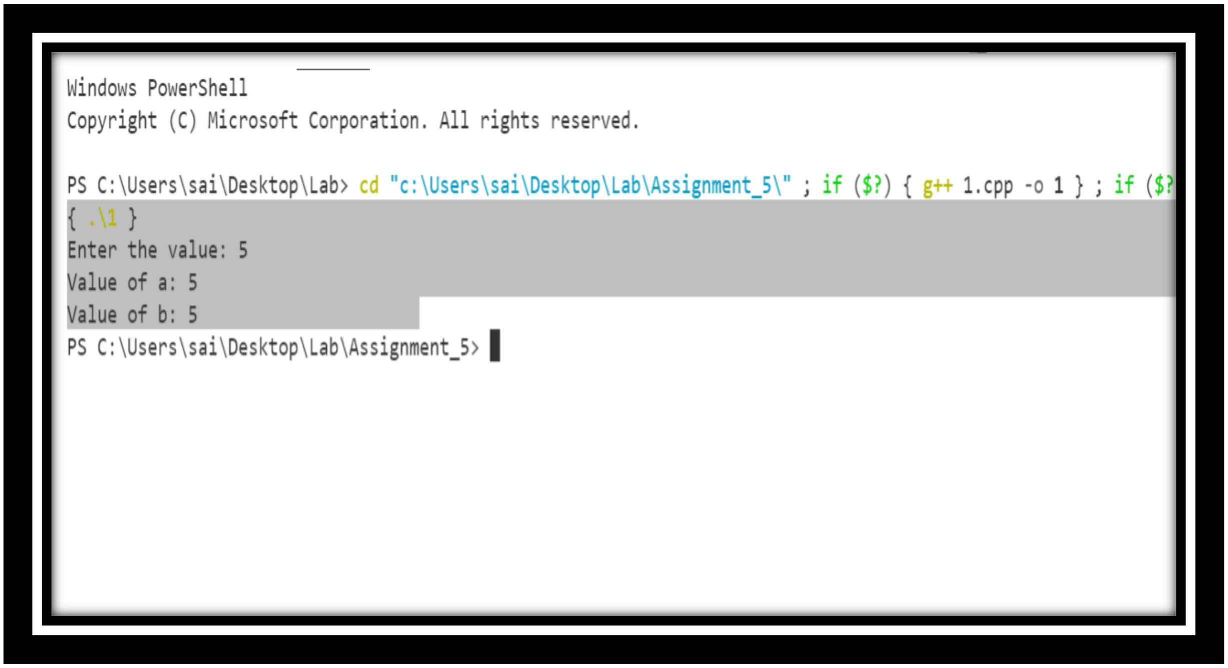
```
PS C:\Users\sai\Desktop\Lab> cd "c:\Users\sai\Desktop\Lab\Assignment_5\" ; if ($?) { g++ 1.cpp -o 1 } ; if ($?)
{ .\1 }
Without using Keywords:
Enter the value of a: 23
Value of a: 23
Enter the value of b: 34
Value of b: 34

Using Keywords:
Enter the value of a: 2
Value of a: 2
Enter the value of b: 34
Value of b: 34
PS C:\Users\sai\Desktop\Lab\Assignment_5>
```

5. Create two namespaces as Outer and Inner, add any one variable, any one function in both the namespaces.
   Write a program to display members of both Outer and Inner namespace.

```
 #include <iostream>
using namespace std;
namespace Outer
{
    int a;
    void get()
    {
        cout << "Value of a: " << a << endl;
    }
}
namespace Inner
{
    int b;
    void get()
    {
        cout << "Value of b: " << b << endl;
    }
}
```

```
int main()
{
    int x;
    cout << "Enter the value: ";
    cin >> x;
    Outer ::a = x;
    Outer ::get();
    Inner ::b = x;
    Inner ::get();
    return 0;
}
```

Output:

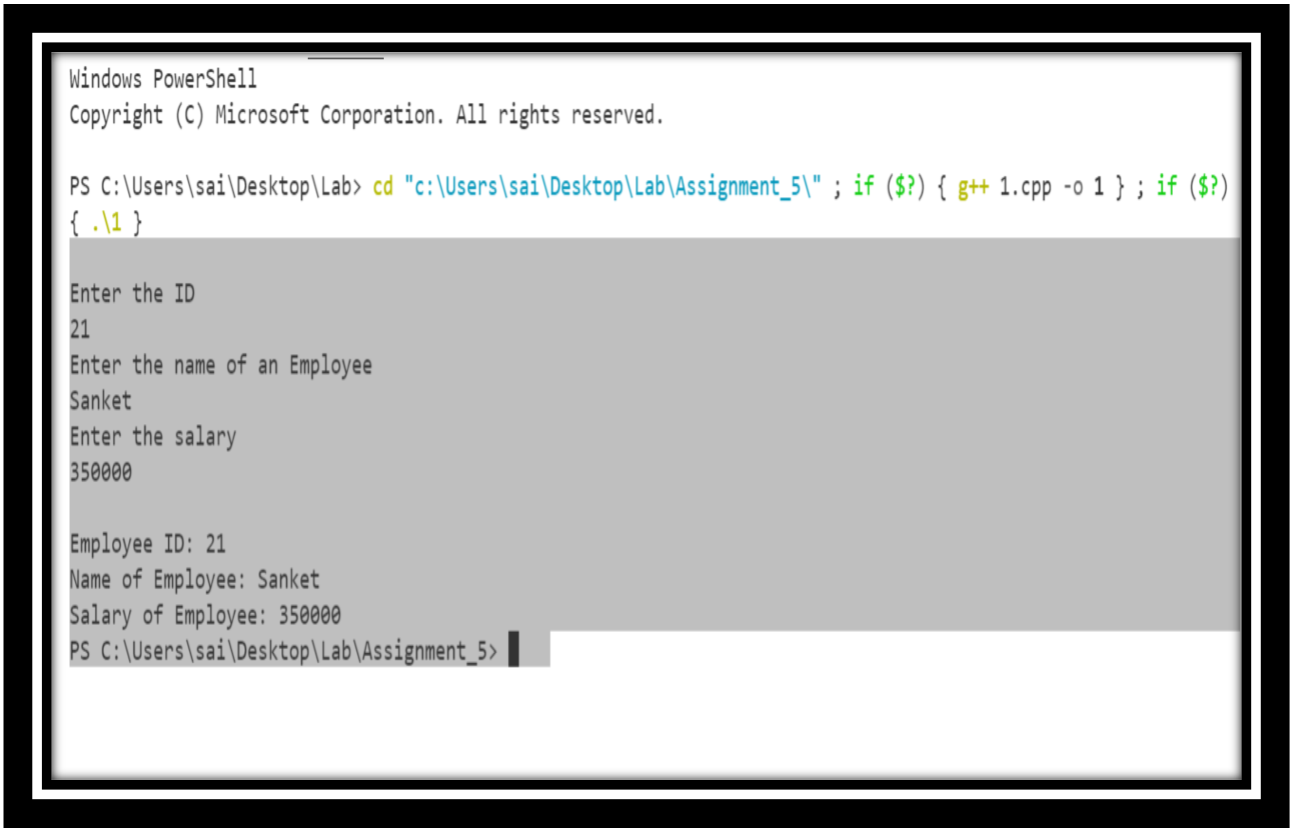


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\sai\Desktop\Lab> cd "c:\Users\sai\Desktop\Lab\Assignment_5\" ; if ($?) { g++ 1.cpp -o 1 } ; if ($?
{ .\1 }
Enter the value: 5
Value of a: 5
Value of b: 5
PS C:\Users\sai\Desktop\Lab\Assignment_5>
```

6. Create a class Employee in namespace Organization. Create functions as setDetails (EmpID, Name, Salary) and getDetails ().
   Write a program to call both the functions using object of Employee class.

```
# include <bits/stdc++.h>
using namespace std;
namespace Organisation{
class Employee{
    int id;
    string name;
    float salary;

    public:
        void set(){
            cout<<"\nEnter the ID\n";
            cin>>id;
            cout<<"Enter the name of an Employee\n";
            cin>>name;
            cout<<"Enter the salary\n";
            cin>>salary;
        }

        void get(){
            cout<<"\nEmployee ID: "<<id<<endl;
            cout<<"Name of Employee: "<<name<<endl;
            cout<<"Salary of Employee: "<<salary<<endl;
        }
};
}
int main(){
    using  namespace Organisation;
    Employee e;
    e.set();
    e.get();
    return 0;
}
```

Output:



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

PS C:\Users\sai\Desktop\Lab> cd "c:\Users\sai\Desktop\Lab\Assignment_5\" ; if ($?) { g++ 1.cpp -o 1 } ; if ($?)
{ .\1 }

Enter the ID
21
Enter the name of an Employee
Sanket
Enter the salary
350000

Employee ID: 21
Name of Employee: Sanket
Salary of Employee: 350000
PS C:\Users\sai\Desktop\Lab\Assignment_5>
```