

LAB-WORK 11

1. Write a C++ program using a Constructor.

CODE:

```
Constructor.cpp > ...
1  //by default every class create a constructor
2  #include <iostream>
3  using namespace std;
4  class Human
5  {
6  protected:
7      double age;
8      double heights;
9  public:
10     Human(){
11         cout<<"This is accessing constructor.\n";
12     }
13 };
14 class man: public Human
15 {
16 public:
17     void print(){
18         cout<<"Enter the age: ";
19         cin>>age;
20         cout<<"Enter the Heights(in inch): ";
21         cin>>heights;
22         cout<<"\n"<<"The Age and Heights(in inch) is: "<<age <<" "<<heights<<endl;
23     }
24 };
25 int main()
26 {
27     man n1;
28     n1.print();
29     cout<<"\n"<<"The program successfully terminated.";
30     return 0;
31 }
```

OUTPUT:

```
This is accessing constructor.
Enter the age: 22
Enter the Heights(in inch): 5.6

The Age and Heights(in inch) is: 22 5.6

The program successfully terminated.
PS E:\Code\CSE 4192\Class\Class-11> █
```

02. Write a C++ program using Destructor.

CODE:

```
Destructor.cpp > main()
1  #include <iostream>
2  using namespace std;
3  class Human
4  {
5  protected:
6      double age;
7      double height;
8  public:
9      Human(){
10         cout<<"Accessing constructor"<<endl;
11     }
12     ~Human(){
13         cout<<"Accessing destructor"<<endl;
14     }
15 };
16 int main()
17 {
18     Human p1;
19     return 0;
20 }
21
```

OUTPUT:

```
Accessing constructor
Accessing destructor
PS E:\Code\CSE 4192\Class\Class-11>
```

03. Write a C++ Program using Copy Constructor.

CODE:

```
G+ Copy_Constructor.cpp > ...
1  #include <iostream>
2  using namespace std;
3  class wall
4  {
5  private:
6      double length;
7      double height;
8  public:
9      wall(double len, double hgt){
10         length= len;
11         height = hgt;
12     }
13     wall(wall &obj){
14         length = obj.length;
15         height = obj.height;
16     }
17     double calculateArea()
18     {
19         return length*height;
20     }
21 };
22 int main()
23 {
24     wall wall1(10.5, 8.6);
25     wall wall2 = wall1;
26     wall wall3 = wall2;
27     cout<<"Area of wall 1: "<<wall1.calculateArea()<<endl;
28     cout<<"Area of wall 2: "<<wall2.calculateArea()<<endl;
29     cout<<"Area of wall 3: "<<wall3.calculateArea()<<endl;
30 }
```

OUTPUT:

```
Area of wall 1: 90.3
Area of wall 2: 90.3
Area of wall 3: 90.3
PS E:\Code\CSE 4192\Class\Class-11>
```

04. Write a C++ program using polymorphism (overloading).

CODE:

```
Polymorphism.cpp > ...
1  //function overloading
2  #include<bits/stdc++.h>
3  using namespace std;
4  class Geeks
5  {
6  public:
7      void func(int x)
8      {
9          cout<<"Value of x is "<<x<<endl;
10     }
11     void func(double x)
12     {
13         cout<<"Value of x is "<<x<<endl;
14     }
15     void func(int x, int y)
16     {
17         cout<<"Value of x and y is "<<x<<" "<<y<<endl;
18     }
19 };
20 int main()
21 {
22     Geeks obj1;
23     obj1.func(7);
24     obj1.func(9.123);
25     obj1.func(85,64);
26     return 0;
27 }
28 |
```

OUTPUT:

```
Value of x is 7
Value of x is 9.123
Value of x and y is 85 64
PS E:\Code\CSE 4192\Class\Class-11>
```

05. Write a C++ program using run time polymorphism.

CODE:

```
run_time_polymorphism.cpp > ...
1  #include<bits/stdc++.h>
2  using namespace std;
3  class animal
4  {
5  public:
6      void eat()
7      {
8          cout<<"Animal Eating..."<<endl;
9      }
10 };
11 class Dog: public animal
12 {
13 public:
14     void eat()
15     {
16         cout<<"Child class Dog Eating bread..."<<endl;
17     }
18 };
19 int main()
20 {
21     Dog d;
22     animal a;
23     //a.eat();
24     d.eat();
25     return 0;
26 }
27
```

OUTPUT:

```
Child class Dog Eating bread...
PS E:\Code\CSE 4192\Class\Class-11>
```