

1.

```
main.cpp X

1 #include <iostream>
2 using namespace std;
3
4 class Shape {
5 public:
6     void display() {
7         cout << "This is a shape" << endl;
8     }
9 }
10
11 class Polygon : public Shape {
12 public:
13     void display() {
14         cout << "Polygon is a shape" << endl;
15     }
16 }
17
18 class Rectangle : public Polygon {
19 public:
20     void display() {
21         cout << "Rectangle is a polygon" << endl;
22     }
23 }
24
25 class Triangle : public Polygon {
26 public:
27     void display() {
28         cout << "Triangle is a polygon" << endl;
29     }
30 }
31
32 class Square : public Rectangle {
33 public:
34     void display() {
35         cout << "Square is a rectangle" << endl;
36     }
37 }
38
39 int main() {
40     Shape s;
41     Polygon p;
42     Rectangle r;
43     Triangle t;
44     Square sq;
45
46     s.display();
47     p.display();
48     r.display();
49     t.display();
50     sq.display();
51
52     return 0;
53 }
54
```

```
□ "C:\Information Systems degr" X + ▾  
This is a shape  
Polygon is a shape  
Rectangle is a polygon  
Triangle is a polygon  
Square is a rectangle  
  
Process returned 0 (0x0) execution time : 0.026 s  
Press any key to continue.
```

2

```
1  include <iostream>
2  include <string>
3  using namespace std;
4
5  static int rollCounter = 1;
6
7  class Marks {
8  protected:
9      int rollNo;
10     string name;
11 public:
12     Marks(string n) {
13         rollNo = rollCounter++;
14         name = n;
15     }
16     virtual int getMarks() = 0; // pure virtual
17     virtual void display() = 0;
18 };
19
20 class Database : public Marks {
21 protected:
22     int dbMarks;
23 public:
24     Database(string n, int m) : Marks(n) {
25         dbMarks = m;
26     }
27     int getMarks() { return dbMarks; }
28     void display() {
29         cout << "Roll: " << rollNo << " | Name: " << name << " | Database Marks: " << dbMarks << endl;
30     }
31 };
32
33 class Statistics : public Marks {
34 protected:
35     int statMarks;
36 public:
37     Statistics(string n, int m) : Marks(n) {
38         statMarks = m;
39     }
40     int getMarks() { return statMarks; }
41     void display() {
42         cout << "Roll: " << rollNo << " | Name: " << name << " | Statistics Marks: " << statMarks << endl;
43     }
44 };
45
46 class DataStructures : public Marks {
47 protected:
48     int dsMarks;
49 public:
50     DataStructures(string n, int m) : Marks(n) {
51         dsMarks = m;
52     }
53     int getMarks() { return dsMarks; }
54     void display() {
55         cout << "Roll: " << rollNo << " | Name: " << name << " | Data Structures Marks: " << dsMarks << endl;
56     }
57 }
```

```
57 L:
58
59 int main() {
60     int n;
61     cout << "Enter number of students: ";
62     cin >> n;
63
64     int total = 0;
65     for(int i = 0; i < n; i++) {
66         string name;
67         int db, st, ds;
68         cout << "Enter name of student " << i+1 << ": ";
69         cin >> name;
70         cout << "Database marks: ";
71         cin >> db;
72         cout << "Statistics marks: ";
73         cin >> st;
74         cout << "Data Structures marks: ";
75         cin >> ds;
76
77         Database d(name, db);
78         Statistics s(name, st);
79         DataStructures daObj(name, ds);
80
81         int studentTotal = d.getMarks() + s.getMarks() + daObj.getMarks();
82         total += studentTotal;
83         cout << "Total marks of " << name << " = " << studentTotal << endl;
84     }
85
86     cout << "Average marks of class = " << (float)total / n << endl;
87     return 0;
88
89
```

```
"C:\Information Systems degr" X + | v
Enter number of students: 300
Enter name of student 1: sandipa
Database marks: 200
Statistics marks: 288
Data Structures marks: 233
Total marks of sandipa = 721
Enter name of student 2: 400
Database marks: 300
Statistics marks: 100
Data Structures marks: 299
Total marks of 400 = 699
Enter name of student 3:
```

```
1 #include<iostream>
2 using namespace std;
3 class base {
4 public:
5     void fun_1() { cout << "base-1\n"; }
6     virtual void fun_2() { cout << "base-2\n"; }
7     virtual void fun_3() { cout << "base-3\n"; }
8     virtual void fun_4() { cout << "base-4\n"; }
9 };
10 class derived : public base {
11 public:
12     void fun_1() { cout << "derived-1\n"; }
13     void fun_2() { cout << "derived-2\n"; }
14     void fun_4(int x) { cout << "derived-4\n"; }
15 };
16 int main() {
17     base *p;
18     derived obj1;
19     p = &obj1;
20     p->fun_1(); // Early binding
21     p->fun_2(); // Late binding
22     p->fun_3(); // Late binding
23     p->fun_4(); // Late binding
24 }
25
```

```
"C:\Information Systems degr" X + | v
base-1
derived-2
base-3
base-4

Process returned 0 (0x0)  execution time : 0.121 s
Press any key to continue.
```

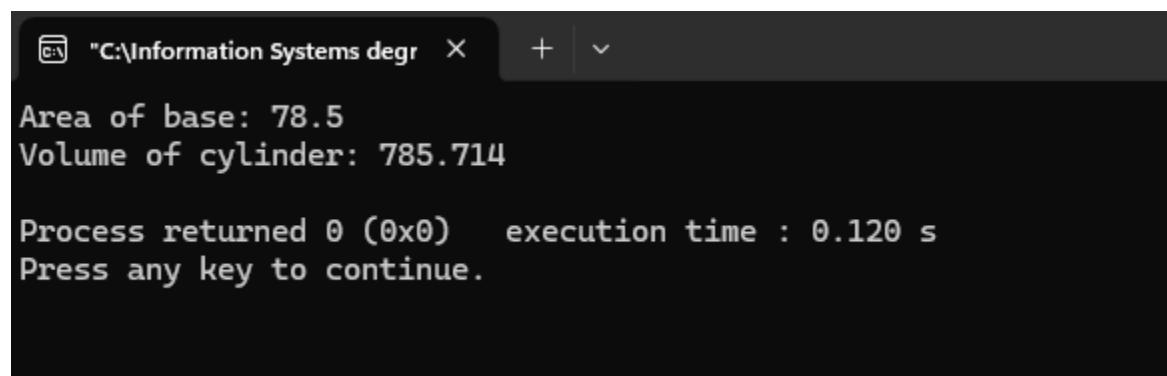
4

```
1 #include <iostream>
2 using namespace std;
3
4 class Sweet {
5 protected:
6     int sweetsCount;
7 public:
8     Sweet(int count) { sweetsCount = count; }
9     int getCount() { return sweetsCount; }
10};
11
12 class Chocolate : public Sweet {
13 public:
14     Chocolate(int count) : Sweet(count) {}
15};
16
17 class IceCream : public Sweet {
18 public:
19     IceCream(int count) : Sweet(count) {}
20};
21
22 int main() {
23     Chocolate c(10);
24     IceCream i(5);
25
26     cout << "Chocolates: " << c.getCount() << endl;
27     cout << "Ice Creams: " << i.getCount() << endl;
28     cout << "Total Sweets: " << c.getCount() + i.getCount() << endl;
29
30     return 0;
31 }
```

```
"C:\Information Systems degr" X + | v
Chocolates: 10
Ice Creams: 5
Total Sweets: 15

Process returned 0 (0x0)  execution time : 0.114 s
Press any key to continue.
```

```
1 #include <iostream>
2 using namespace std;
3
4 class Circle {
5 protected:
6     float radius;
7 public:
8     void setRadius(float r) { radius = r; }
9 };
10
11 class Rectangle {
12 protected:
13     float length, breadth;
14 public:
15     void setDimensions(float l, float b) { length = l; breadth = b; }
16 };
17
18 class Cylinder : public Circle, public Rectangle {
19 public:
20     float area() {
21         return 3.14 * radius * radius;
22     }
23     float volume() {
24         return (22.0/7) * radius * radius * length;
25     }
26 };
27
28 int main() {
29     Cylinder cyl;
30     cyl.setRadius(5);
31     cyl.setDimensions(10, 0); // breadth not needed
32
33     cout << "Area of base: " << cyl.area() << endl;
34     cout << "Volume of cylinder: " << cyl.volume() << endl;
35
36     return 0;
37 }
```



```
C:\> "C:\Information Systems degr" + | v
Area of base: 78.5
Volume of cylinder: 785.714

Process returned 0 (0x0)  execution time : 0.120 s
Press any key to continue.
```