

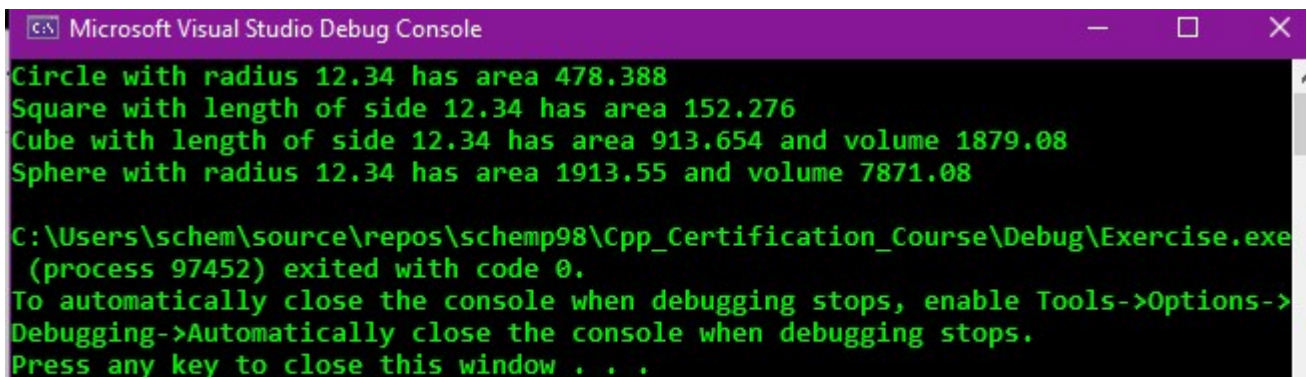
```
1  //
2  // Shaun Chemplavil U08713628
3  // shaun.chemplavil@gmail.com
4  // C/C++ Programming III : Intermediate Programming with Objects
5  // 151116 Raymond L. Mitchell III
6  // Complex.cpp
7  // Win10
8  // Visual C++ 19.0
9  //
10 // File contains the member functions for Circle, Square, Sphere and Cube class
11 //
12
13 #include <iostream>
14 #include "Shape.h"
15
16 using std::cout;
17
18 const double PI = 3.141592653589793238462;
19
20 using ShaunChemplavil::Shape;
21 using ShaunChemplavil::TwoDimensionalShape;
22 using ShaunChemplavil::ThreeDimensionalShape;
23
24 using ShaunChemplavil::Circle;
25 using ShaunChemplavil::Square;
26 using ShaunChemplavil::Sphere;
27 using ShaunChemplavil::Cube;
28
29 // Default Constructor
30 Circle::Circle(double radius)
31     : radius(radius) {}
32
33 double Circle::getArea() const
34 {
35     return (radius * radius * PI);
36 }
37
38 void Circle::display() const
39 {
40     cout << "Circle with radius " << radius << " has area " << getArea() << "\n";
41 }
42
43 // Default Constructor
44 Square::Square(double lengthOfSide)
45     : lengthOfSide(lengthOfSide) {}
46
47 double Square::getArea() const
48 {
49     return (lengthOfSide * lengthOfSide);
50 }
51
52 void Square::display() const
```

```
53     {
54         cout << "Square with length of side " << lengthOfSide
55             << " has area " << getArea() << "\n";
56     }
57
58
59     Sphere::Sphere(double radius)
60         : radius(radius) {}
61
62     double Sphere::getSurfaceArea() const
63     {
64         return (4.0 * PI * radius * radius);
65     }
66
67
68     double Sphere::getVolume() const
69     {
70         return (getSurfaceArea() *radius/3.0);
71     }
72
73     void Sphere::display() const
74     {
75         cout << "Sphere with radius " << radius<< " has area "
76             << getSurfaceArea() << " and volume " << getVolume() << "\n";
77     }
78
79     // Default Constructor
80     Cube::Cube(double lengthOfSide)
81         : lengthOfSide(lengthOfSide) {}
82
83     double Cube::getSurfaceArea() const
84     {
85         return (lengthOfSide * lengthOfSide*6.0);
86     }
87
88     double Cube::getVolume() const
89     {
90         return (lengthOfSide * lengthOfSide * lengthOfSide);
91     }
92     void Cube::display() const
93     {
94         cout << "Cube with length of side " << lengthOfSide << " has area " <<
95             getSurfaceArea()
96             << " and volume " << getVolume() << "\n";
97     }
98
```

```
1 //
2 // Shaun Chemplavil U08713628
3 // shaun.chemplavil@gmail.com
4 // C/C++ Programming III : Intermediate Programming with Objects
5 // 151116 Raymond L. Mitchell III
6 // Complex.cpp
7 // Win10
8 // Visual C++ 19.0
9 //
10 // File contains the member functions for Circle, Square, Sphere and Cube class
11 //
12
13 #include <iostream>
14 #include "Shape.h"
15
16 using std::cout;
17
18 const double PI = 3.141592653589793238462;
19
20 using ShaunChemplavil::Shape;
21 using ShaunChemplavil::TwoDimensionalShape;
22 using ShaunChemplavil::ThreeDimensionalShape;
23
24 using ShaunChemplavil::Circle;
25 using ShaunChemplavil::Square;
26 using ShaunChemplavil::Sphere;
27 using ShaunChemplavil::Cube;
28
29 // Default Constructor
30 Circle::Circle(double radius)
31     : radius(radius) {}
32
33 double Circle::getArea() const
34 {
35     // area = pi*radius^2
36     return (radius * radius * PI);
37 }
38
39 void Circle::display() const
40 {
41     cout << "Circle with radius " << radius << " has area " << getArea() << "\n";
42 }
43
44 // Default Constructor
45 Square::Square(double lengthOfSide)
46     : lengthOfSide(lengthOfSide) {}
47
48 double Square::getArea() const
49 {
50     // area = lengthOfSide^2
51     return (lengthOfSide * lengthOfSide);
52 }
```

```
53
54 void Square::display() const
55 {
56     cout << "Square with length of side " << lengthOfSide
57         << " has area " << getArea() << "\n";
58 }
59
60 Sphere::Sphere(double radius)
61     : radius(radius) {}
62
63 double Sphere::getSurfaceArea() const
64 {
65     // surface area = 4*pi*r^2
66     return (4.0 * PI * radius * radius);
67 }
68
69 double Sphere::getVolume() const
70 {
71     // volume = 4/3*pi*r^3
72     return (getSurfaceArea() * radius / 3.0);
73 }
74
75 void Sphere::display() const
76 {
77     cout << "Sphere with radius " << radius << " has area "
78         << getSurfaceArea() << " and volume " << getVolume() << "\n";
79 }
80
81 // Default Constructor
82 Cube::Cube(double lengthOfSide)
83     : lengthOfSide(lengthOfSide) {}
84
85 double Cube::getSurfaceArea() const
86 {
87     // surface area = 6*lengthofSide^2
88     return (lengthOfSide * lengthOfSide * 6.0);
89 }
90
91 double Cube::getVolume() const
92 { // surface area = lengthofSide^3
93     return (lengthOfSide * lengthOfSide * lengthOfSide);
94 }
95 void Cube::display() const
96 {
97     cout << "Cube with length of side " << lengthOfSide << " has area " << 
98         << getSurfaceArea()
99         << " and volume " << getVolume() << "\n";
100 }
```

```
1 //
2 // Shaun Chemplavil U08713628
3 // shaun.chemplavil@gmail.com
4 // C/C++ Programming III : Intermediate Programming with Objects
5 // 151116 Raymond L. Mitchell III
6 // hw5.cpp
7 // Win10
8 // Visual C++ 19.0
9 //
10 // Test Program for the Shape class
11 //
12
13 #include "Shape.h"
14
15 using ShaunChemplavil::Circle;
16 using ShaunChemplavil::Square;
17 using ShaunChemplavil::Sphere;
18 using ShaunChemplavil::Cube;
19 using ShaunChemplavil::Shape;
20
21 int main()
22 {
23     Shape * shapePtrArray[4];
24
25     // arbitrary input
26     double shapeInput = 12.34;
27
28     // set up array of pointers to shapes to exercise virtual functions
29     shapePtrArray[0] = new Circle(shapeInput);
30     shapePtrArray[1] = new Square(shapeInput);
31     shapePtrArray[2] = new Cube(shapeInput);
32     shapePtrArray[3] = new Sphere(shapeInput);
33
34     for (int shapeIdx = 0; shapeIdx < 4; shapeIdx++)
35     {
36         // call display virtual function via call by reference
37         shapePtrArray[shapeIdx]->display();
38         // Clean up dynamically allocated memory (utilizes virtual destructors)
39         delete shapePtrArray[shapeIdx];
40     }
41 }
42
```



Microsoft Visual Studio Debug Console

Circle with radius 12.34 has area 478.388
Square with length of side 12.34 has area 152.276
Cube with length of side 12.34 has area 913.654 and volume 1879.08
Sphere with radius 12.34 has area 1913.55 and volume 7871.08

C:\Users\schem\source\repos\schemp98\Cpp_Certification_Course\Debug\Exercise.exe
(process 97452) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->
Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .