**Program 13-1**

1 // This program demonstrates a simple class.

2 #include <iostream>

3 using namespace std;

4

5 // Rectangle class declaration.

6 class Rectangle

7 {

8 private:

9 double width;

10 double length;

11 public:

12 void setWidth(double);

13 void setLength(double);

14 double getWidth() const;

15 double getLength() const;

16 double getArea() const;

17 };

18

19 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

20 // setWidth assigns a value to the width member. \*

21 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

22

23 void Rectangle::setWidth(double w)

24 {

25 width = w;

26 }

27

28 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

29 // setLength assigns a value to the length member. \*

30 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

31

32 void Rectangle::setLength(double len)

33 {

34 length = len;

35 }

36

37 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

38 // getWidth returns the value in the width member. \*

39 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

40

41 double Rectangle::getWidth() const

42 {

43 return width;

44 }

45

46 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

47 // getLength returns the value in the length member. \*

48 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

49

50 double Rectangle::getLength() const

51 {

52 return length;

53 }

54

55 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

56 // getArea returns the product of width times length. \*

57 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

58

59 double Rectangle::getArea() const

60 {

61 return width \* length;

62 }

63

64 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

65 // Function main \*

66 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

67

68 int main()

69 {

70 Rectangle box; // Define an instance of the Rectangle class

71 double rectWidth; // Local variable for width

72 double rectLength; // Local variable for length

73

74 // Get the rectangle's width and length from the user.

75 cout << "This program will calculate the area of a\n";

76 cout << "rectangle. What is the width? ";

77 cin >> rectWidth;

78 cout << "What is the length? ";

79 cin >> rectLength;

80

81 // Store the width and length of the rectangle

82 // in the box object.

83 box.setWidth(rectWidth);

84 box.setLength(rectLength);

85

86 // Display the rectangle's data.

87 cout << "Here is the rectangle's data:\n";

88 cout << "Width: " << box.getWidth() << endl;

89 cout << "Length: " << box.getLength() << endl;

90 cout << "Area: " << box.getArea() << endl;

91 return 0;

92 }

**Program Output with Example Input Shown in Bold**

This program will calculate the area of a

rectangle. What is the width? **10 [Enter]**

What is the length? **5 [Enter]**

Here is the rectangle's data:

Width: 10

Length: 5

Area: 50

**Program 13-10**

1 // This program demonstrates a destructor.

2 #include <iostream>

3 using namespace std;

4

5 class Demo

6 {

7 public:

8 Demo(); // Constructor

9 ~Demo(); // Destructor

10 };

11

12 Demo::Demo()

13 {

14 cout << "Welcome to the constructor!\n";

15 }

16

17 Demo::~Demo()

18 {

19 cout << "The destructor is now running.\n";

20 }

21

22 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

23 // Function main. \*

24 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

25

26 int main()

27 {

28 Demo demoObject; // Define a demo object;

29

30 cout << "This program demonstrates an object\n";

31 cout << "with a constructor and destructor.\n";

32 return 0;

33 }

**Program Output**

Welcome to the constructor!

This program demonstrates an object

with a constructor and destructor.

The destructor is now running.

**Contents of InventoryItem.h**

1 // This class has overloaded constructors.

2 #ifndef INVENTORYITEM\_H

3 #define INVENTORYITEM\_H

4 #include <string>

5 using namespace std;

6

7 class InventoryItem

8 {

9 private:

10 string description; // The item description

11 double cost; // The item cost

12 int units; // Number of units on hand

13 public:

14 // Constructor #1

15 InventoryItem()

16 { // Initialize description, cost, and units.

17 description = "";

18 cost = 0.0;

19 units = 0; }

20

21 // Constructor #2

22 InventoryItem(string desc)

23 { // Assign the value to description.

24 description = desc;

25

26 // Initialize cost and units.

27 cost = 0.0;

28 units = 0; }

29

30 // Constructor #3

31 InventoryItem(string desc, double c, int u)

32 { // Assign values to description, cost, and units.

33 description = desc;

34 cost = c;

35 units = u; }

36

37 // Mutator functions

38 void setDescription(string d)

39 { description = d; }

40

41 void setCost(double c)

42 { cost = c; }

43

44 void setUnits(int u)

45 { units = u; }

46

47 // Accessor functions

48 string getDescription() const

49 { return description; }

50

51 double getCost() const

52 { return cost; }

53

54 int getUnits() const

55 { return units; }

56 };

57 #endif

**Program 13-12**

1 // This program demonstrates a class with overloaded constructors.

2 #include <iostream>

3 #include <iomanip>

4 #include "InventoryItem.h"

5

6 int main()

7 {

8 // Create an InventoryItem object and call

9 // the default constructor.

10 InventoryItem item1;

11 item1.setDescription("Hammer"); // Set the description

12 item1.setCost(6.95); // Set the cost

13 item1.setUnits(12); // Set the units

14

15 // Create an InventoryItem object and call

16 // constructor #2.

17 InventoryItem item2("Pliers");

18

19 // Create an InventoryItem object and call

20 // constructor #3.

21 InventoryItem item3("Wrench", 8.75, 20);

22

23 cout << "The following items are in inventory:\n";

24 cout << setprecision(2) << fixed << showpoint;

25

26 // Display the data for item 1.

27 cout << "Description: " << item1.getDescription() << endl;

28 cout << "Cost: $" << item1.getCost() << endl;

29 cout << "Units on Hand: " << item1.getUnits() << endl << endl;

30

31 // Display the data for item 2.

32 cout << "Description: " << item2.getDescription() << endl;

33 cout << "Cost: $" << item2.getCost() << endl;

34 cout << "Units on Hand: " << item2.getUnits() << endl << endl;

35

36 // Display the data for item 3.

37 cout << "Description: " << item3.getDescription() << endl;

38 cout << "Cost: $" << item3.getCost() << endl;

39 cout << "Units on Hand: " << item3.getUnits() << endl;

40 return 0;

41 }

**Program Output**

The following items are in inventory:

Description: Hammer

Cost: $6.95

Units on Hand: 12

Description: Pliers

Cost: $0.00

Units on Hand: 0

Description: Wrench

Cost: $8.75

Units on Hand: 20