

5.8 How functions work

Students:
Section 5.9 is a part of 2 assignments: **CSC108 CH05.8-5.16 C5B** ▾
Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. [Learn more](#)

Includes: CA
Due: 04/15/2025, 11:59 PM EDT

5.9 Functions: Common errors

A common error is to copy-and-paste code among functions but then not complete all necessary modifications to the pasted code. For example, a programmer might have developed and tested a function to convert a temperature value in Celsius to Fahrenheit, and then copied and modified the original function into a new function to convert Fahrenheit to Celsius as shown:

Figure 5.9.1: Copy-paste common error: Pasted code not properly modified. Find error on the right.

```
double Cel2Fah(double celVal) {  
    double convTmp;  
    double fahVal;  
  
    convTmp = (9.0 / 5.0) * celVal;  
    fahVal = convTmp + 32;  
    return fahVal;  
}  
  
double Fah2Cel(double fahVal) {  
    double convTmp;  
    double celVal;  
  
    convTmp = fahVal - 32;  
    celVal = convTmp * (5.0 / 9.0);  
    return celVal;  
}
```

[Feedback?](#)

The programmer forgot to change the return statement to return celVal rather than fahVal. Copying-and-pasting code is a common and useful time-saver, and can reduce errors by starting with known-correct code. Our advice is that when you copy-paste code, be extremely vigilant in making all necessary modifications. Just as the awareness that dark alleys or wet roads may be dangerous can cause you to vigilantly observe your surroundings or drive carefully, the awareness that copying-and-pasting is a common source of errors, may cause you to more vigilantly ensure you modify a pasted function correctly.

PARTICIPATION ACTIVITY 5.9.1: Copy-pasted sum-of-squares code.

Original parameters were num1, num2, num3. Original code was:

```
int sum;  
sum = (num1 * num1) + (num2 * num2) + (num3 * num3);  
return sum;
```

New parameters are num1, num2, num3, num4. Find the error in the copy-pasted new code below.

```
1) int sum;  
  
sum =  
(num1 * num1) + (num2 * num2) + (num3 * num3) + (num3 * num4);  
;  
return sum;
```

[Feedback?](#)

Another common error is to return the wrong variable, such as typing `return convTmp;` instead of `fahVal` or `celVal`. The function will work and sometimes even return the correct value.

Failing to return a value for a function is another common error. If execution reaches the end of a function's statements, the function automatically returns. For a function with a void return type, such an automatic return poses no problem, although some programmers recommend including a return statement for clarity. But for a function defined to return a value, the returned value is undefined; the value could be anything. For example, the user-defined function below lacks a return statement:

Figure 5.9.2: Missing return statement common error: Program may sometimes work, leading to hard-to-find bug.

```
#include <iostream>  
using namespace std;  
  
int StepsToFeet(int baseSteps) {  
    const int FEET_PER_STEP = 3; // Unit conversion  
    int feetTot; // Corresponding feet to steps  
  
    feetTot = baseSteps * FEET_PER_STEP;  
}  
  
int main() {  
    int stepsInput; // User defined steps  
    int feetTot; // Corresponding feet to steps  
  
    // Prompt user for input  
    cout << "Enter number of steps walked: ";  
    cin >> stepsInput;  
  
    // Call functions to convert steps to feet  
    feetTot = StepsToFeet(stepsInput);  
    cout << "Feet: " << feetTot << endl;  
  
    return 0;  
}
```

Enter number of steps walked: 1000
Feet: 3000

[Feedback?](#)

Sometimes a function with a missing return statement (or just `return;`) still returns the correct value. The reason is that the compiler uses a memory location to return a value to the calling expression. That location may have also been used by the compiler to store a local variable of that function. If that local variable happens to be the item that was supposed to be returned, the value in that location is the correct return value. But a later seemingly unrelated change to a function, like defining a new variable, may cause the compiler to use different memory locations, and the function suddenly no longer returns the correct value, leading to a bewildered programmer.

PARTICIPATION ACTIVITY 5.9.2: Common function errors.

Find the error in the function's code.

```
1) int ComputeSumOfSquares(int num1, int num2) {  
    int sum;  
  
    sum =  
(num1 * num1) + (num2 * num2);  
;  
    return;  
}
```

```
2) int ComputeEquation1(int num, int val, int k) {  
    int sum;  
  
    sum = (num * val) + (k * val);  
;  
    return num;  
}
```

[Feedback?](#)

PARTICIPATION ACTIVITY 5.9.3: Common function errors.

1) Forgetting to return a value from a function is a common error.

- True
- False

2) Copying-and-pasting code can lead to common errors if all necessary changes are not made to the pasted code.

- True
- False

3) Returning the incorrect variable from a function is a common error.

- True
- False

4) Is this function correct for squaring an integer?

```
int sgr(int a) {  
    int t;  
    t = a * a;  
}
```

- Yes
- No

5) Is this function correct for squaring an integer?

```
int sgr(int a) {  
    int t;  
    t = a * a;  
    return a;  
}
```

- Yes
- No

[Feedback?](#)

CHALLENGE ACTIVITY 5.9.1: Function errors: Copying one function to create another.

Using the CelsiusToKelvin function as a guide, create a new function, changing the name to KelvinToCelsius, and modifying the function accordingly.

[Learn how our auto grader works](#)

```
620890.5010016.q3zay7  
1 #include <iostream>  
2 using namespace std;  
3  
4 double CelsiusToKelvin(double valueCelsius) {  
5     double valueKelvin;  
6  
7     valueKelvin = valueCelsius + 273.15;  
8  
9     return valueKelvin;  
10 }  
11 /* Your solution goes here */  
12  
13 int main() {  
14     double valueC;  
15     double valueK;  
16  
17     valueC = 10.0;  
18 }
```

[Run](#)

[Feedback?](#)

How was this section? [Provide section feedback](#)

Activity summary for assignment: **CSC108 CH05.8-5.16 C5B** ▾

0 / 19 points

Due: 04/15/2025, 11:59 PM EDT

Please browse to this assignment through BlackboardLearn so zyBooks knows where to send your activity. [Learn more](#)

Completion details ▾

[Feedback?](#)

5.10 Pass by reference