Notes #6.1

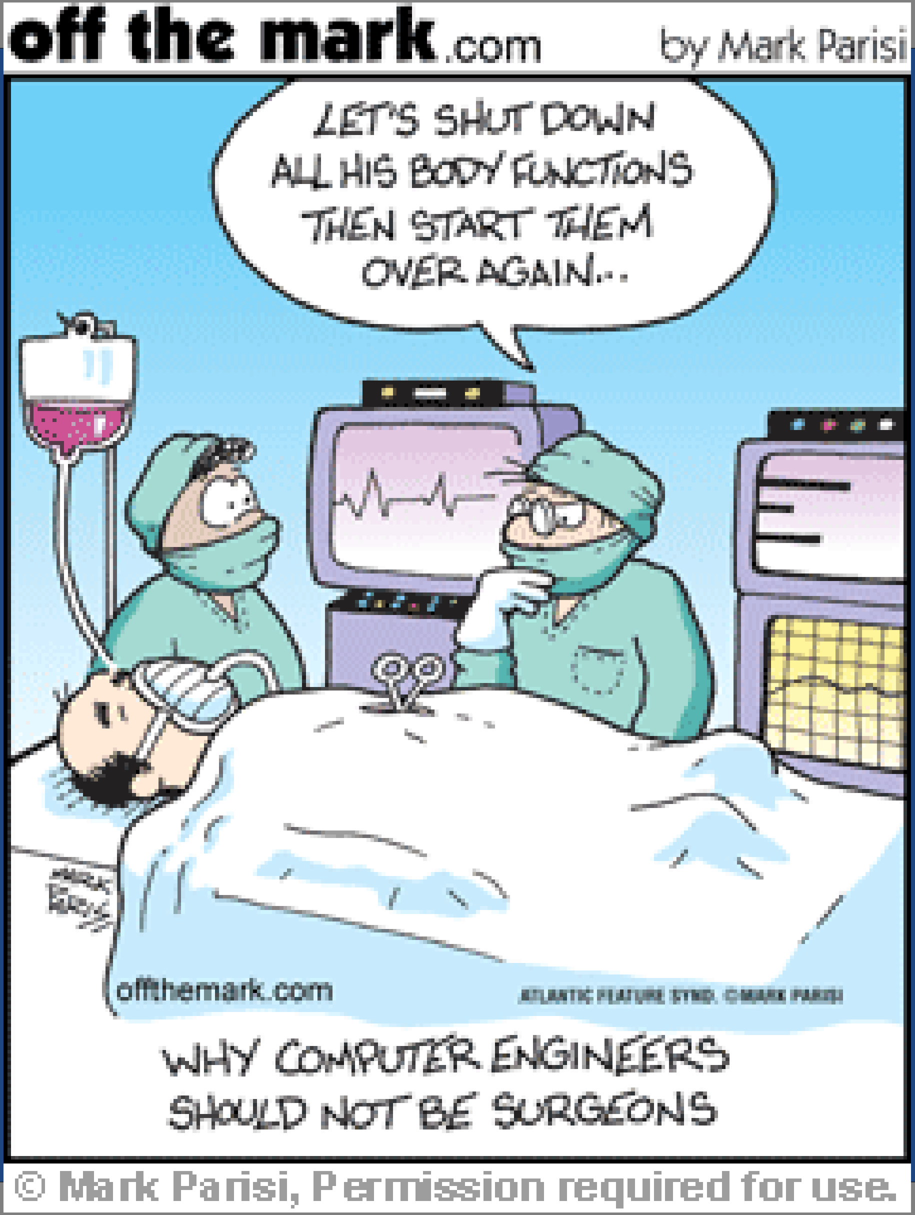
Functions

Part 1

Chapter 9

CMPT 130

© Dr. B. Fraser


off the mark.com
by Mark Parisi
LET'S SHUT DOWN ALL HIS BODY FUNCTIONS THEN START THEM OVER AGAIN...
offthemark.com
ATLANTIC FEATURE SYND. ©MARK PARISI
WHY COMPUTER ENGINEERS SHOULD NOT BE SURGEONS
© Mark Parisi, Permission required for use.

Topics

1) How can we break up a program into smaller sections?
2) How can we pass information to and from functions?
3) How long do variables exist for? (next week)

Functions

Functions

- Functions:
  - Each function should perform...
  - Also called methods, or procedures.
  - Ex:
    - calculate a value, display the menu.
  - Allows for the divide and conquer approach:
    - Divide: split the big problem down into multiple smaller problems.
    - Conquer:..

Function definition

```
// A simple C++ program.
#include <iostream>
using namespace std;

void displayMsg( )
{
    cout << "Hello world\n";
}

int main( )
{
    displayMsg();
    return 0;
}
```

The type of value/information the function returns.

displayMsg.

List of variables to hold values passed into the function.

Statements to carry out the task of the function.

Must have () on call or..

Function definition

- A function (like a variable) must be..
  - For the moment, put the definition of a function earlier (above) in the file than any calls to the function; otherwise will not compile.
- Function Return Type:
  - a specific type (such as int or bool or char); or
  -

Review

- What is the difference between defining a function and calling a function?

- Write a function to display "I code therefore I am."

Getting data in and out of a function.

Function Parameters

Function call (use):

Arguments

```
int main( )
{
    displayNTimes("hi", 5);
    return 0;
}
```

Function definition:

Parameter List

```
void displayNTimes(string msg, int n) {
    while (n > 0) {
        cout << msg << endl;
        n--;
    }
}
```

Parameters

- **Arguments:** ..
- **Parameter List:** ..
  - Inside the (...) of the function header.
  - May be empty if no parameters required.
- **Parameters:** ..
  - These are variables inside the method.

Returning a value

- The return statements does 2 things:
  - Causes the current function to exit, returning control to the calling function.
  -

```
/*
    Return the number of points the user scored based
    on the number of zombies killed.
    Returns 0 if number killed is less than 0.
*/
int calcScore(int numZombies)
{
    if (numZombies < 0) {
        return 0;
    }
    return numZombies * POINTS_PER_ZOMBIE;
}
```

Returning a value vs Printing a value

- When a function calculates a value, it usually..

- Analogy:
  - You are voting in a referendum on a mail-in ballot, mailed to you by Elections Canada.
  - Do you say your vote aloud, or return your ballot to Elections Canada?

```
//...
int getVote()
{
    return 1;
}
```

```
//...
void getVote()
{
    cout << 1 << endl;
}
```

Review

- Write a function:
  - named add()
  - which accepts 2 int parameters; and
  - returns the sum of the two parameters as an int