COMP1917: 06 Functions

Sim Mautner

s.mautner@unsw.edu.au

August 7, 2016

Resources

- Moffat, Chapter 5: Getting Started with Functions

Programming Language Principles

Four techniques provided by almost all programming languages:

- Calculation: doing arithmetic to compute new values
- Selection: choosing between alternative execution paths
- Iteration: repeating a computation until desired conditions are met
- Abstraction: creating units which can be reused, and whose internal details are hidden from outside inspection.

Abstraction via Functions

Functions allow you to:

- separate out “encapsulate” a piece of code serving a single purpose
- test and verify a piece of code
- reuse the code
- shorten code resulting in easier modification and debugging

Functions we already use:

- From stdio.h: `printf()`, `scanf()`
- From stdlib.h: `rand()`

Structure of a Function

1. Return type
2. Function name
3. Parameters (inside brackets, comma separated)
4. Return statement

```
int addNumbers(int num1, int num2) { // 1, 2, 3

   int sum = num1 + num2;
   return sum; // 4

}
```

Functions with No Return Value

1. Return type: void
2. No return statement necessary.

```
void printAsterisks(int numAsterisks) {

   int i=0;
   while(i < numAsterisks) {
      printf("*");
      i++;
   }
   printf("\n");

}
```

Function Prototypes

- Each function has a function prototype.
- It tells the compiler that the function exists, and the structure it has.
- It includes the key information about the function.
- Examples:

```
int addNumbers(int num1, int num2);
void printAsterisks(int numAsterisks);
```

Program Structure

1. Header comment
2. #included files
3. #defines
4. prototypes
5. main function
6. functions

For more information see the Style Guide:
https://wiki.cse.unsw.edu.au/info/CoreCourses/StyleGuide

Noteworthy Features

- Each function can have 0 or more parameters.
- Each function can only return 0 values, or a single value.
- Each function stores its own local copy of the parameters passed to it. The original version of the variables remain unaltered.
- Parameters received by the function, and local variables created by the function, are all discarded when the function returns.

Try It Yourself

- Choose some programs written in previous tutes, labs and lectures and change them so that they are written using one or more functions.