CS 110 SI Session 8

Functions Review, Pass by Reference
Session Outline

1. Quick Review
   1.1 Pass by Value Functions
   1.2 Pass by Reference Functions
   1.3 Extracting Functions
2. Examples
Functions

• As you saw last week in the lab, you can create your own functions

• What is a function?
  – A small, self contained block of code which can be declared once and then run from different parts of your program.
Why are functions useful?

- Avoid repeated code
- Reusable among different programs
- Can be tested individually
- Divide and conquer complex algorithms
3 Steps to Creating a Function

1. Declare it (write the body after main)

2. Prototype it (copy the signature before main)

3. Use it (call it from main)
#include <iostream> // program continued
using namespace std;
void printHello();
void printHello();
int main () {
  printHello();
  return 0;
} // continued right
Pass by Value

• In most functions you write, the function will not get the original variable, but instead a copy of the variable.
  – This is called “pass by value.”

• This keeps the original data safe in case the function goes rogue and wrecks the data.
Pass by value function example

```cpp
#include <iostream>
using namespace std;

int timesTen(int input);

int main () {
    cout << timesTen(3);
    return 0;
}

// program continued
int timesTen(int input) {
    input = input * 10;
    return input;
}
```
Pass by Reference

- When you trust your function completely, you can choose to share the exact same variables with the function that you have with main.
  - This is known as Pass by Reference
  - Rather than the function getting a copy of the variable, it will get the exact same variable, even though it may refer to it as another name.
  - Pass by reference parameters have the “&” sign attached to the variable name or data type.
Pass by Reference

• Allows you to...
  
  – Share the same memory space between your main program and your subfunctions

  – Get back multiple outputs from your function
Pass by reference function example

```cpp
#include <iostream>
using namespace std;

void timesTen(int &input);

int main () {
    int x = 3;
    timesTen(x);
    cout << x << endl;
    return 0;
}

// program continued

void timesTen
    (int &input) {
    input = input * 10;
}
```
Pass by Reference

• This tends to be one of the more mystifying concepts in CS 110.
  – Helps to trace memory usage by drawing boxes

• Remember: even though your sub function refers to the variable by its own unique name, it is still the same memory location. If you change one the other will change.

• The following examples may help clarify this...
What does the following code print?

```cpp
int doubleIt (int x);
int main () {
    int y = 3;
    cout << doubleIt(y);
    cout << y;
}
int doubleIt (int x) {
    x = x * 2;
    return x;
}
```
What does the following code print?

```cpp
int doubleIt (int x);
int main () {
    int y = 3;
    cout << doubleIt(y); //6
    cout << y; //3
}
int doubleIt (int x) {
    x = x * 2;
    return x;
}
```
void swapValues (int x, int &y, int &z);

int main () {
    int x = 3;
    int y = 7;
    int z = 2;
    modifyValues (x, y, z);
    cout << x << " " << y << " " << z << endl;
}

void swapValues (int x, int &y, int &z) {
    x = y; y = z; z = x;
}
Examples

• Let's work through some new examples and some code refactoring examples so that this stuff sticks.
New Example - Counting Stars

Write a function which takes a string, and returns the count of the character "*" within that string.

myString = "abc**edf*qrtop";
countStars (myString); // returns 3.
Extracting Functions

• When learning to use functions, sometimes it is difficult to think of where they would apply

• A good idea is to go back to programs that you have written, and look for places where functions could be substituted in for code you have done

• This is known as “Extraction Functions”, a form of Code Refactoring.
Refactoring Example

- For the following examples we have already created c++ solutions.
- Can you improve those solutions by implementing parts of the program as functions?
Password Validation

• A website has a requirement that a password must contain:
  – 2 numeric digits
  – 2 Uppercase letters

• Write a program that allows the user to enter a string, and then checks if the string meets the password requirements for the website.
Refactor the code so that...

- It uses one function
- It takes the password
- It tells the user if the password is valid
- Ask:
  - What data types need to come in
  - What data types need to come out
  - Does the function need to share the variables with main or just the values?
Write a program that allows a user to enter a string of lowercase letters and a encryption number.

Shift every character in the string ahead by the key number.

If a letter exceeds 'z', roll it over to 'a' and keep advancing.

Print the encrypted string.
Refactor the code so that it...

- Uses a function which takes the unencrypted string and a key, and gives back the encrypted string.

- Do you want the original data to change?
Find the closest point

- A text file named “coords.txt” contains a set of coordinates in the format
  
x  y
  
  eg.
  
  12  4
  8  7

- Write a program which finds the point that is closest to the origin.
Refactor the code so that it...

- Uses two functions
- One function deals with File I/O.
- One function computes distance.