Notes #6.2

Functions

Part 2

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? (Part 1)
2) How can we pass information to and from functions? (Part 1)
3) How long do variables exist for?

Local vs Global Scope

Local variables

- Local variables:
  Variable declared inside a function.
  - Restricted scope (visibility) to within the function.
  - Restricted lifetime to when function is executing.
  - (These Includes function parameters.)
- What's that mean?
  - Cannot use a local variable outside the function.
  - Local variables are...
    destroyed when function ends.
    Next time through, a new one is created.

Global variables

- Global variables are..
  declared outside of all functions.
  - Accessible anywhere between its definition and the end of the .cpp file.
  - Lifetime is the same as the program.
- Guidelines:
  - Good for constants:
    const int DAYS_PER_WEEK = 7;
  - Often problematic for variables: can be very..
    hard to understand and debug global variables.
    - Use local variables as much as possible.

Scope and variable names

- Scope is ..
  the part of the program where a variable can be accessed.
  - Global scope: ..outside of all other scopes.
  - Local scope to a function: Inside a function.
  - Blocks: Any block {...}, such as for a while loop.
- You *could* reuse a variable name in different nested scopes, but is very confusing!
  - Try and give variables in nested scope unique names.

Scope and Lifetime

Scope

		Local	Global
Lifetime	Temporary	Local variable	
	Persistent	Static local variable	Global variable

Scope, Functions and Variable Names

- Functions have their own scope
  - Therefore functions can contain variables with the same names ... remember *main* is a function
    - This applies to both parameters and to variables declared inside a function
    - Variables with the same name in different scopes are different variables
- Reusing variable names in nested scope is generally a bad idea
  - But it is often acceptable to reuse variable names in different functions

~ scopeName.cpp

Practise Review Questions

- Write just function headings (no body) for the following :
  - apple(): takes two ints, returns a float.
  - orange(): takes two ints and prints out the sum.
- Write a function named max() which:
  - Accepts two int values
  - Returns the maximum of the two.
- Write a function named range() which:
  - Accepts two char parameters.
  - Prints all characters between (and including) the input two characters.
  - Prints “ERROR” if the second char is < the first char.

Summary

- Function definition: type, name, parameter list, body.
- Function call must use (): int age = getAge();
- Use return to pass back a value.
- Scope
  - Local variables exist only inside the function.
  - Global variables often bad; global constants good