Pre-Defined Functions in C++
Review for Midterm #1

CS 16: Solving Problems with Computers I
Lecture #6

Ziad Matni
Dept. of Computer Science, UCSB

Announcements

- **Homework #5 due today**
- Homework #6 issued today is due NEXT THURSDAY
  - Because next Tuesday is midterm#1 exam
- Lab #3 is due Monday, 5/1

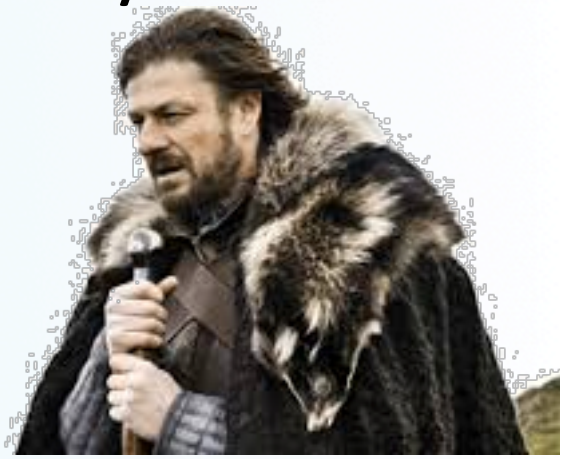
- Your grades are now online!

  Access them through the class website
  and click on "**Class Grades, CMPSC 16, Spring 2017**"

- **Don't forget your TAs' and Instructor's office hours!!** ☺

MIDTERM IS COMING!

- Material: ***Everything*** we've done**, incl. up to Th. 4/20**
  - Homework, Labs, Lectures, Textbook
- **Tuesday, 4/25** in this classroom
- **Starts at 12:30pm **SHARP** (come early)**
- **Pre-assigned seating**
- Duration: **1 hour long**



- Closed book: no calculators, no phones, no computers
- Only 1 sheet (single-sided) of written notes
  - Must be no bigger than 8.5" x 11"
  - **You have to turn it in with the exam**
- **You will write your answers on the exam sheet itself.**

Bring your UCSB IDs to the exam!!!

What's on the Midterm#1?
From the Lectures, including…

- Intro to Computers, Programming, and C++
- Variables and Assignments
- Boolean Expressions (comparison of variables)
- Input and Output on Standard Devices (cout, cin)
- Data Types, Escape Sequences, Formatting Decimal
- Arithmetic Operations and their Priorities
- Flow of Control & Conditional Statements
- Boolean Logic Operators
- Loops: for, while, do while
- Types of Errors in Programming
- Multiway Branching and the switch command
- Command Line Inputs to C++ Programs
- Functions in C++
- Generating Random Numbers

What's on the Midterm#1?
Readings from the Textbook

- Ch. 1
- Ch. 2
- Ch. 3
- Ch. 4
  - Section 4.2, 4.3, 4.4, 4.5

What's on the Midterm#1?
From the Assignments and Labs

- Review homework questions

- Review labs and understand what you did
  - The lab processes and experiences, especially
  - The process of writing a program and compiling it
  - The basic UNIX commands you use in lab:
    - cd, ls, g++

Sample Question
Multiple Choice

Complete the following C++ code that is supposed to print the numbers 2, 3, 4, 5, 6:

```
int c = 0;
while (_____________) {
 cout << c+2 << " ";
 c++; }
```

A. c < 7
B. c > 5
C. (c + 2) < 6
D. (c + 2) != 6
E. c < 5

Sample Question
Multiple Choice

What is the exact output of this C++ code?

```
int prod(1);
for (int m = 1; m <= 9; m++) {
   prod *= m;
   m += 2;
}
cout << "Total product is: " << prod << endl;
```

A. Total product is: 720
B. Total product is: 120
C. Total product is: 28
D. Total product is: 2
E. Total product is: 1

Sample Question
Short-Answer Coding

Write C++ code showing a function definition of **`distance( )`** which takes 4 **int** values x_1, x_0, y_1, and y_0 and returns a **double** data type that's equal to

$$\sqrt{(x_1-x_0)^2 + (y_1-y_0)^2}\,.$$

Assume that the **cmath** lib has been imported.

```
double distance(int x1, int x0, int y1, int y0)
{
        double a = pow(x1 – x0, 2);
        double b = pow(y1 – y0, 2);
        double z = sqrt(a + b);
        return z;
}
```

Note:
When I ask for "code", that means not a complete program.

Otherwise I'd ask for a "program". Also, this would be clear from the question.

Sample Question

Coding Syntax: ***Find The 10 Mistakes***

```
1   #include <iostream>
2   #include <stringer>
3   using namepaces std;
4
5   int main () {
6     int number; x = 0;
7     string word;
8
9     cout << "Enter an integer: /n";
10    cin >> number
11    cout << "Enter a string: \n";
12    cin << word;
13
14    while (x < number);
15    {
16       cout << words << " ";
17       x+++;
18    }
19    cout >> endl; return 0;
20  }
```

2: Should be: **<string>**

3: Should be: **using namespace std;**

6: Should be: **int number, x = 0;**

9: Should be: **\n**

10: Missing **;** at the end

11: Should be: **cin >> word;**

14: Must remove the **;** at the end

16: Should be: **cout << word << " ";**

17: Should be: **x++**

19: Should be: **cout << endl; return 0;**

Example of a Simple Function in C++

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

int sum2nums(int num1, int num2);

int main ( ) {
    int a(3), b(5);
    int sum = sum2nums(a, b);
    cout << sum << endl;
    return 0;
}

int sum2nums(int num1, int num2) {
    return (num1 + num2);
}
```

Predefined Functions in C++

- C++ comes with "built-in" libraries of predefined functions
- Example: sqrt function (found in the library *cmath*)
  - Computes and returns the square root of a number

```
the_root = sqrt(9.0);
```

  - The number 9 is called *the argument*
  - After calculation, the variable **the_root** will be equal to 3.0
- Can variable **the_root** be either int or double?

Other Predefined **cmath** Functions

- pow(x, y) --- **double** value = pow(2, -8);
  - Returns 2^{-8} , a double value (0.00390625)
  - Arguments are of type double

- sin(x), cos(x), tan(x), etc… --- **double** value = sin(1.5708);
  - Returns sin($\pi/2$) (equal to 1) – note it's in radians
  - Argument is of type double

Other Predefined **cmath** Functions

- abs(x) --- **int** value = abs(-8);
  - Returns absolute value of argument x
  - Return value is of type **int**
  - Argument is of type int

- fabs(x) --- **double** value = fabs(–8.0);
  - Also returns absolute value of argument x
  - Return value is of type **double**
  - Argument is of type double

Random Number Generation: Step 1

- Not true-random, but pseudo-random numbers.

  Must
  ```
  #include <cstdlib>
  #include <ctime>
  ```

- First, ***seed*** the random number generator only once

  ```
  srand(time(0)); //place inside main( )
  ```

  – **time( )** is a pre-defined function in the **ctime** library
  (it gives the current system time)

  – It's used here because it generates a *distinctive enough seed*, so that **rand( )** generates a "good enough" random number.

Random Number Generation: Step 2

- Next, use the **rand( )** function, which returns a random integer that is greater than or equal to 0 and less than RAND_MAX
  (a library-dependent value, but is at least 32767)

```
int r = rand();
```

- But what if you want to generate random numbers in other ranges? Example, between 1 and 6?

Random Numbers

- Use **%** and **+** to scale to the number range you want

- For example to get a random number bounded from 1 to 6 to simulate rolling a six-sided die:

```
int die = (rand( ) % 6) + 1;
```

What Will This Program Print Out?

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

int main ( ) {
    srand(time(0));
    int throw_times, die;
    cout << “How many times shall we throw the die?!\n”;
    cin >> throw_times;
    for (int i=0; i < throw_times; i++) {
            die = (rand( ) % 6) + 1;
            cout << “We threw a ” << die << endl; }
    return 0;
}
```

Type Casting

- Recall the problem with integer division in C++:

```
int total_candy = 9, number_of_people = 4;
double candy_per_person;
candy_per_person = total_candy / number_of_people;
```

  - candy_per_person = 2, not 2.25!

- A **Type Cast** produces a value of one type ***from*** another
  - **static_cast<double>(total_candy)**
    produces a double representing
    the integer value of total_candy

Type Cast Example

```
int total_candy = 9, number_of_people = 4;
double candy_per_person;
candy_per_person =
    static_cast<double>(total_candy)/number_of_people;
```

- **candy_per_person now is 2.25!**

- The following would also work:

```
candy_per_person =
    total_candy / static_cast<double>(number_of_people);
```

- This, however, would not!

```
candy_per_person = static_cast<double>
                        (total_candy / number_of_people);
```

Integer division occurs before type cast!

Question

- Can you determine the value of d?

double d = 11 / 2;

Integer division occurs before type cast!

- What about this value of d?

double d = 11.0 / 2.0;

TO DOs

- Finish reading Chapter 4 for Thursday's class
- Finish Homework6 (due Thursday 4/27)
- Finish Lab3 (due Monday 5/1)
- Prep for Lab4 next Wednesday (due Tuesday 5/2)

- **Study for Midterm #1!!!!** ☺
- Come see the prof. or the TAs during office hours if you have questions

</LECTURE>