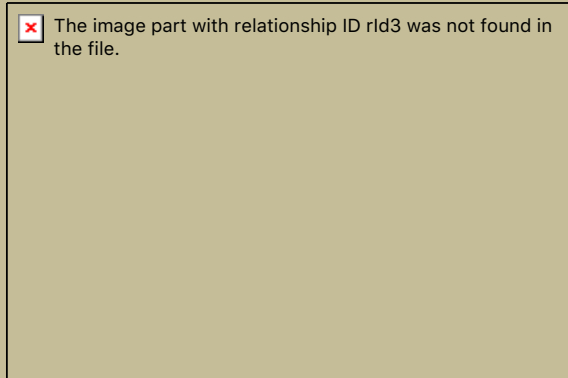


COMPUTER SCIENCE 1: STARTING COMPUTING CSCI 1300



Ioana Fleming / Vipra Gupta
Spring 2018
Lecture 12



University of Colorado
Boulder

Agenda

- This week
 - for loops
 - arrays



Announcements

- Rec 5 due on 2/17
- Hmwk 4 (1st Project) due 2/18
- Practicum 1 has been scheduled:
 - February 21st, 2018
 - In lecture. 50 minutes. Bring a laptop!
 - Review: in lecture on 2/19
 - Visible in *Tentative Schedule* on Moodle



Here is what will be on the Practicum

- Multiple Choice Questions (4 or 5 questions)
- Write 3 functions to solve 3 tasks
- Tasks will require you apply
 - Declare functions (return value, parameters)
 - Declare and assign values to variables
 - Use boolean expression for conditionals
 - IF statements
 - IF-ELSE statements
 - Nested IF statements (IF-ELSE IF - ...)
 - Iteration via a WHILE statement



Practicum - logistics

- Bring your student ID (knowing your number is not enough)
- 50 minutes
- No cheat sheet
- No communication: no phones, smart watches, messaging apps, no windows open except Moodle and Cloud9
- You have access to all previous created solutions in Cloud9



Practice, practice, practice!

- Review all previous Moodle programming questions from previous recitation and homework assignments
- Review examples we did in class
- Time is short; prepare accordingly.
- Two Practice Practicum Quizzes:
 - one with programming questions
 - one with multiple-choice questions



Tips for Timed Exam

- Read the Questions
 - read them not once, but **TWICE** before starting the code
 - follow all the instructions explicitly (especially for names and order of parameters)
- Create or Modify a Code
 - know your C++ syntax
- Create and Use an IF, IF ELSE
 - know your C++ syntax
 - know how to create a condition
- Create and Use a WHILE
 - know your C++ syntax
 - know how to iterate through a string's characters
- Passing parameters
 - know your C++ syntax



for Loop Syntax

for (Init_Action; Bool_Exp; Update_Action)
 Body_Statement

- Like if-else, Body_Statement can be a block statement
 - Much more typical



for Loop Example

- `for (count=0; count<3; count++)`
`{`
`cout << "Hi "; // Loop Body`
`}`
- How many times does loop body execute?
- Initialization, loop condition and update all "built into" the for-loop structure!
- A natural "counting" loop

While loops vs for loops

- Cloud9 examples: *liftoff.cpp*



Loop Issues

- Loop's condition expression can be ANY boolean expression
- Examples:

```
while (count<3 && done!=0)
{
    // Do something
}
```

```
for (index=0;index<10 && entry!=-99)
{
    // Do something
}
```



Loop Pitfalls: Misplaced ;

- Watch the misplaced ; (semicolon)

- Example:

```
while (response != 0) ; ←  
{  
    cout << "Enter val: ";  
    cin >> response;  
}
```

- Notice the ";" after the while condition!

- Result here: INFINITE LOOP!



Loop Pitfalls: Infinite Loops

- Loop condition must evaluate to false at some iteration through loop
 - If not → infinite loop.
 - Example:

```
while (1)
{
    cout << "Hello ";
}
```
 - A perfectly legal C++ loop → **always infinite!**
- Can we create an infinite *for* loop?



Loop Pitfalls: Infinite Loops

- Can we create an infinite *for* loop?
- In the case of *for* loops, if we modify the loop variable

- Example:

```
for (i = 0; i < 5; i++)  
{  
    cout << "Hello ";  
    i--;  
}
```



Practice Problem

Read string from the user. If string is email address display “You have mail!”

- looking for the @ character



Nested Loops

- Recall: ANY valid C++ statements can be inside body of loop
- This includes additional loop statements!
 - Called "nested loops"

- Requires careful indenting:

```
for (outer=0; outer<5; outer++)  
    for (inner=7; inner>2; inner--)  
        cout << outer << inner;
```

- Notice no { } since each body is one statement
- Good style dictates we use { } anyway

