Exam 1 Review

CS 1428
Fall 2017

Jill Seaman

Exam 1

- Thursday, Oct 5
- In class, closed book, closed notes, clean desk
- 15% of your final grade
- 80 minutes to complete it
- Bring your ID card!!!!
- Bring a number 2 pencil! (and eraser)
- NO: calculators or cell phones.
- NO: headphones/earbuds.

Exam Format

- 100 Points total
  - 50 points: 25 multiple choice and T/F (scantron form)
  - 50 points: writing code on the test paper

    ➡ programs and individual statements
- Tasks:
  - Tracing code (what is the output)
  - Finding errors in code
  - Evaluating C++ expressions
  - Demonstrate general knowledge about C++ and programming
  - Programming (writing code)

Content from Textbook

Units 1 through 3:

- Chapter 1: 1.1-3
- Chapter 2: 2.1-17 (except 2.11)
- Chapter 3: 3.1-10
- Chapter 4: 4.1-15 (except 4.13)
- Chapter 5: 5.1 and 5.11 (first half)

See reading list online for specific topics of each section

Intro to Computers and Programming

- Definitions: Computer, Program, Programmer
- Hardware vs Software
- Hardware components: (cpu, main memory, secondary storage, input and output devices)
- Program vs. Algorithm
- Programming languages: machine lang vs low level lang vs high level lang
- Compilation: source code file -> executable
- Execution

Introduction to C++

- Literals: numbers, characters, strings
  - Special characters
- Identifiers, rules for valid names
- Variable Definitions and Initialization
- Assignment Statements
- Data Types
  - int, short, long, float, double, bool, char, string
  - values/ranges (rough idea)
  - suitability of each for various types of data
- Scope rules, comments, named constants

Expressions and Types

- Numerical Expressions
  - Operators: +, -, *, /, % (modulus)
  - Precedence rules, parens ( )
- Type Conversions:
  - binary operations
  - assignment
  - explicit type casting
- Integer division vs float division
- Pow(a,b) and other Math library functions

Assignment operators

- Multiple assignment
  - `a = b = c = 4;`
- Combined Assignment operators
  - `+=  -=  *=  /=`
- Increment and Decrement
  - `x++  y--`
- Hand Tracing a program

Input and output

- cout, stream insertion operator (<<), endl
- cin, stream extraction operator (>>)
- formatting: setw, setprecision, fixed, left/right
- inputting characters and strings
  - >> vs getline(cin,var)
  - using cin >> ws to solve problem of >> followed by getline
- using file stream objects for file I/O:
  - using ifstream, ofstream variables
  - open and close, << and >>

Ifs and boolean expressions

- Relational and Logical Expressions
  - Rel. Operators: < <= > >= == !=
  - Logical Operators: ! && ||
  - Precedence rules, parens
- if statements:
  - if
  - if-else
  - nested if statements
  - if-else if (reformatting of nested if statements)
  - block or compound statement

Switch Statements
and programming with conditions

- Input validation
- Comparing characters and strings
- The switch statement
  - the break statement
  - switch case fall-through, multiple labels
- Menus
- Scope of variables in blocks

Sample problem: what is output?

- What is the output of the following statements?

```
int fox = 6;
float dog = 5.7;
dog = fox + dog;
if (fox > dog)
   cout << "Hello!";
else if (fox < dog)
   cout << dog;
else
   cout << fox;
cout << endl;
cout << fixed << setprecision(1);
cout << "dog is: " << dog << endl;
```

A)
```
Hello!
dog is: 5.7
```

B)
```
Hello!
dog is: 11.7
```

C)
```
11.7
dog is: 11.7
```

D)
```
6
dog is: 5.7
```

Sample problem: Programming

- Write a C++ program that reads a floating point number representing the side of a square from a file named “test.txt” and outputs the area of the square formatted to five decimal places. If the area is greater than 1,000, it should also output the following statement:

```
That’s really big!
```


How to study

- Review the slides (these, and Units 1 - 3)
  - understand all the concepts, **quiz yourself**
- Use the book to help understand the slides
  - there will be no questions over material that is in the book but not on the slides
- Review programming assignments (fix yours!)
  - get printouts of solutions 2 and 3 up front or in my office
- Review the Top Hat questions
- Try some exercises from the book
- Practice, practice, practice! Write code! Sleep!