

Exam 2 Review

CS 1428
Fall 2014

Jill Seaman

1

Exam 2

- Wednesday, November 12
- In class, closed book, closed notes, clean desk
- 15% of your final grade
- 80 minutes to complete it
- I recommend using a pencil (and eraser)
- All writing will be done on the test paper I will hand out.
- No calculators or cell phones.

2

Exam Format

- 100 Points total
 - ▶ 30-35 points: writing programs/functions
 - ▶ Multiple choice
 - ▶ Short answers
 - ▶ Short code segments
- Tasks:
 - ▶ Tracing code (what is the output)
 - ▶ Finding errors in code
 - ▶ Label parts of a program/functions
 - ▶ Programming (writing code)

3

Content from Textbook

- Chapter 4: 4.10-15 (except 4.13)
- Chapter 5: 5.2-12
- Chapter 6: 6.1-5, 7-10, and 13
- Chapter 7: 7.1-6 (except 7.5)

Weeks 5 through 10, primarily loops, arrays, and functions

4

Switch Statements and programming with conditions

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

5

Loops

- while loop
 - ▶ general purpose
- do-while
 - ▶ body always done once
 - ▶ good for menus, repeating a process
- for loop
 - ▶ init; test; update
 - ▶ all are optional
- Which loops are good for which situations

6

Loops

- Using a while loop for input validation
- Counters/count controlled loop
- Keeping a running total
- Sentinel controlled loop
- Nested loops
- Reading data from a file of unknown length
 - ▶ while (fin >> number)
- Break and continue
- Infinite loops

7

Arrays

- Array declaration/definition:
 - ▶ `int list[10];`
 - ▶ size declarator limitation (in the C++ standard)
- Array elements
 - ▶ `list[i]`
 - ▶ range of subscripts
 - ▶ types
- Array initialization:
 - ▶ `int list[] = {6,7,8};`

8

Arrays

- Processing arrays
 - input and output
 - sum, average
 - finding max, min (and index of which one)
 - counting values that pass a test
 - array assignment (copy)
 - array compare (for equality)
- Partially filled arrays
- Lack of bounds checking

9

Functions

- Function definition
 - name, return type, parameter list, body
- Function call
 - name, argument list
- Function prototype, when it is required
- Function parameters and arguments
 - Understand how they work

10

Functions

- The return statement
 - returning a value from a function
 - calling a function that returns a value
- Pass by value
- Pass by reference
- Scope
 - local and global variables
 - parameters
 - scope and lifetime
 - global constants

11

Sample problem: what is output?

Show the EXACT output of the following program:

```
int list[] = {8,10,3,55,1,2,3,7};
int x=10;
int i = 3;

while (i < 8) {
    x++;
    int t = list[i];
    if (t < 10) {
        x = list[i+1];
    } else if (t < 20) {
        x++;
    } else {
        x--;
    }
    i = i+3;
    cout << "x = " << x << endl;
}
}
```

12

Sample problem: Programming

The formula for converting a temperature from Fahrenheit to Celsius is

$$C = \frac{5}{9}(F - 32)$$

where F is the Fahrenheit temperature and C is the Celsius temperature. Write a function named `celsius` that accepts a Fahrenheit temperature as an argument. The function should return the temperature, converted to Celsius. Demonstrate the function by calling it in a loop that displays a table of the Fahrenheit temperatures 0 through 20 and their Celsius equivalents.

13

How to study

- Review the slides (these, and weeks 5 - 10)
 - ▶ 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 4, 5 and 6 up front or in my office
- Try some exercises from the book
- Practice, practice, practice! Write code!
- Get some sleep

14