

CSCI 1320 (Principles of Programming I), Fall 2011

Homework 2

Credit: 30 points.

1 Reading

Be sure you have read chapters 3 and 4. (You should be able to do the first two problems with only the material in chapter 3, however.)

2 Programming Problems

Do the following programming problems. You will end up with at least one code file per problem. Submit your program source (and any other needed files) by sending mail to `bmassing@cs.trinity.edu`, with each file as an attachment. Please use a subject line that mentions the course number/name and the assignment (e.g., “csci 1320 homework 2”). You can develop your programs on any system that provides the needed functionality, but I will test them on one of the department’s Linux machines, so you should probably make sure they work in that environment before turning them in.

1. (5 points) Write a Scala program to show the binary representation of an integer. The program should ask the user for an integer and print the integer and its binary equivalent. Here are some sample executions, assuming you called your program `show-binary.scala`. Text in **boldface** is what you type; text in *typewriter* font is what the program prints.

```
[bmassing@xena02]$ scala show-binary.scala
enter an integer:
10
you entered 10
in binary that is 1010
```

```
[bmassing@xena02]$ scala show-binary.scala
enter an integer:
-10
you entered -10
in binary that is 1111111111111111111111111111111110110
```

For extra credit (up to 2 points), make it always print exactly 32 digits.

Hint: This program can be very short if you take advantage of a method mentioned in chapter 3.

2. (5 points) Write a Scala program to convert a Fahrenheit temperature to Celsius. The rule for converting Fahrenheit temperature F to Celsius temperature C is

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

The program should ask the user for the Fahrenheit temperature and print the equivalent Celsius temperature. You can use integers or floating-point numbers for this problem.

3. (10 points) Write a Scala program to compute U.S. income tax given taxable income. The instructions for Form 1040 for 2010 are as follows (for single taxpayers):
- If taxable income is not more than \$100,000, tax is to be looked up in the tax table.
 - If taxable income is more than \$100,000 but not more than \$171,850, tax is 28% of taxable income, minus \$6,290.75.
 - If taxable income is more than \$171,850 but not more than \$373,650, tax is 33% of taxable income, minus \$14,883.25.
 - If taxable income is more than \$373,650, tax is 35% of taxable income, minus \$22,356.25.

Your program should first prompt the user for a taxable income (in whole dollars), and then do one of two things: If the amount is no more than \$100,000, it should tell the user to use the IRS's tax table; otherwise it should use the above formulas to compute and print the tax due, rounded to the nearest dollar (round 50 cents upward). For example, the tax on \$110,000 is 28% of \$110,000 minus \$6,290.75; computing and rounding, this gives \$24,509. (Don't worry about printing amounts with commas; for this assignment it's fine to just print, for the example, \$24509.)

You could use floating-point numbers for the calculation, but the result will likely be more accurate if you instead do everything with integers, calculating tax in pennies and converting back to dollars at the end. (An easy way to round cents to the nearest dollar is to add 50 cents and then use integer division to convert to dollars.)

4. (10 points) Write a Scala program asks the user for three integers and prints them in order from smallest to largest. (Sounds simple — but be sure you consider all possible cases. The user could enter the numbers from smallest to largest, or vice versa, or in several other orders. Your mission is to give the right answer for all possibilities.) *Clarification:* Notice that the numbers do not have to be distinct — for example, the user could enter three values of 0.