

CS 100 Assignment 1: Jump Right In

Summer 2001

Due in lecture, Thursday June 28

In the end we can never be given knowledge by others; we can only be stimulated.
We must develop our own knowledge. -- Charles T. Tart

1. Objectives

Completing all tasks in this assignment will help you:

- locate information and course policies on the webpage
- gain experience using the course newsgroup
- practice creating a project, editing files, and running programs in CodeWarrior

First, skim the whole assignment. Then carefully read all instructions before starting. You must use a **fixed-width** font (e.g., Courier or Monaco) for your homework.

2. Course Information and Newsgroup (10 points)

The course newsgroup is an important resource. It provides a forum for everyone in the class to ask questions and discuss the answers. If you have a question whose answer would be of benefit to the class at large, this is the place to ask it. Likewise, if you believe you know the answer to a question someone else has asked, feel free to post a response. The teaching assistants will be monitoring the newsgroup, but we definitely encourage students to take an active role in contributing answers as well.

Before posting to the newsgroup, make sure you are familiar with the netiquette guidelines found on the Newsgroup page, off the course website.

The newsgroup is **cornell.class.cs100j** (the "j" is for "Java"). You can use any newsreader you like to access the newsgroup. See the webpage for details about how to access it:
<http://www.cs.cornell.edu/Courses/cs100/2001su/newsgroup.html>

Part A:

1. Find the article called "CS 100 Assignment 1" and save it to a file.
2. Edit the file with any text editor or word processor.
3. Write the following information at the top of the file:
Name: (your name)
ID: (your Cornell ID)
Date: (current date)
Assignment: 1, Part A
4. **Answer the questions** (most answers can be found somewhere on the website) inside this file by writing your responses after each question.
5. Save your work. Make sure it is in ASCII text format, if your word processor uses something else as its default.
6. Print your file and turn it in at lecture.

The purpose of this part of the assignment is to make sure you know how to access the newsgroup. However, the CS 100 newsgroup is only one of many, many newsgroups (as you may have noticed from the huge list on WinVN)! We encourage you to explore these other groups. There are newsgroups created for virtually every interest. Take a look at the list (Netscape can also provide such a list) and browse through the ones that interest you. Have fun!¹

¹ Warning: too much time spent reading newsgroups may negatively impact your free time.

3. CodeWarrior (10 points correctness, 10 points style)

In this section, you'll create a project in CodeWarrior and experiment with making changes to a program.

Part B:

1. First, read the Case Study in Savitch, pages 73 – 77, and do the Self-test questions (you do not need to turn these in).
2. Create a new text file with following information at the top of the file:
Name: (your name)
ID: (your Cornell ID)
Date: (current date)
Assignment: 1, Part B
3. Go to the CS 100 website and download **ChangeMaker.java** from the Assignments page (this file can also be found on the Savitch CD). Run the code (create a new project, add this file to the project, add **SavitchIn.java** (can be found on the website) set the Target to ChangeMaker, compile, and run. For more details on how to do this, see the CodeWarrior Guide on the Java page off the website).
4. Play with the ChangeMaker program until you feel comfortable with how it works. Experiment with giving the program different input to see what it does. Can you "break" the program by giving it invalid input or causing it to give the wrong output? **Write a short paragraph in your text file** describing your experiments and any recommendations you would make to improve the program (e.g., ways to fix any problems you identified, better input, better output).
5. **Answer the following questions** in your text file:
 - a) Assuming an input from 1 to 99, what is the largest number of dimes the program could possibly give as output?
 - b) The original code includes the following snippet:

```
dimes = amount/10;  
amount = amount%10;  
nickels = amount/5;  
amount = amount%5;
```

Let's assume that it was instead written in the following way:

```
nickels = amount/5;  
amount = amount%5;  
dimes = amount/10;  
amount = amount%10;
```

Show the output of the original program and for this modified version when given each of the following inputs: **10, 11, 37, 55, 99** (You do not have to type in these modifications, but it may help you determine what the output will be)

Describe the effect of this swap. How did it change the program's behavior?

6. **Create a header** for ChangeMaker.java (using comments) that includes your identifying information from step 1.
7. **Make the following changes** to ChangeMaker.java. Identify your changes by adding **comments** in the appropriate places.
 - The original code offers quarters, dimes, nickels, and pennies in change. Add in the appropriate code to support the possibility of giving **half-dollars** as well (in the same spirit as the original program, your revised version should prefer to give a half-dollar over giving two quarters). Don't forget to add code that includes half-dollars in the output as well.

- Modify the code so that it only outputs lines that involve **non-zero** numbers of coins. For example, if the input is 25, then the output should be
25 cents in coins can be given as:
1 quarters

If the input is 15, then the output should be
15 cents in coins can be given as:
1 dimes
1 nickels

Note that you should remove the "and" that is currently output on the nickels line. If you want to improve the output by causing the program to output "and" where appropriate (the second-to-last line), feel free (this is worth one **Bonus Point**). If not, just delete the "and".

8. Print out your text file and modified ChangeMaker.java and hand them in at lecture.

4. Algorithms and pseudocode (20 points)

Programming is sometimes referred to as *automated problem solving*. A program tells a computer what steps to follow in a language the computer understands. The first step, for the programmer, is to come up with an *algorithm*, which is a high-level list of instructions for how to solve the problem. The algorithm is usually written as a series of steps in regular language (readable by humans); this is sometimes referred to as *pseudocode*. Later, the programmer translates this high-level description into a program in a programming language.

Part C:

1. Make sure you've done Exercise 1 before starting this part of the assignment.
2. Create a new text file with following information at the top of the file:
Name: (your name)
ID: (your Cornell ID)
Date: (current date)
Assignment: 1, Part C
3. **Describe the following task** in algorithm form, print your file, and turn it in at lecture.

Using at most 1 page of single-spaced, 12-point **fixed-width** font, write an algorithm that the instructor could follow in lecture to determine which state the largest group of students are from (e.g., "25 students are from Texas."). Although this algorithm is for a human, not a computer, you should still strive for clarity and precision. Also, try to make your algorithm as efficient as possible. Algorithms with insufficient detail (e.g., "1. Find the state with the largest number of students.") will receive minimal credit.