# Project: Introduction to Java



## Deliverables

Your project files should be submitted to Web-CAT by the due date and time specified. In order to avoid a late penalty for the project, you must submit your completed code files to Web-CAT by 11:59 p.m. on the due date. You may submit your project up to 24 hours after the due date, but there is a late penalty of 15 points. No projects will be accepted after the one-day late period. If you are unable to submit via Web-CAT, you should e-mail your project Java files in a zip file to your TA before the deadline.

Files to submit to Web-CAT:

- MyLifeGoals.java
- JLetter.java

## Specifications

**Overview**: You will write two programs this week. One will print your short-term, medium-term, and long-term life goals to standard output, and the other will display letter J as a large block letter. Because these are small programs, you will only need to have 1-2 sentences of description in your class and method Javadoc comments (don't forget the @author and @version tags in the class-level comment in each file).

- **MyLifeGoals.java**

  **Requirements**: Write the application MyLifeGoals that prints your name, your short-term, medium-term and long-term life goals.

  **Design**: Your program should contain a main method that prints the information listed under "Output" (i.e., your output should replace the text in Italics).

| Output: |
| --- |
| *Your first and last name (separated by a space)*<br>*(The second line should be blank)*<br>*Describe your short-term life goals.(in one line)*<br>*Describe your medium-term life goals. (in one line)*<br>*Describe your long-term life goals. (in one line)* |

  Describe your life goals (if you have never thought about them, this is a good chance to think about them carefully). The actual output for each line of goals should be at least 100 characters not including spaces. Page 1 of Activity 1 shows how to break up a String literal in your program (where lines cannot exceed 80 characters) in order to print a longer line.

**Code and Test**: The expected output for the program will vary from student to student, but it is important to follow output pattern described above, formatting requirements, and minimum character requirements. If you are not sure how many characters (no spaces) are in your output, you can copy your output into Microsoft Word then select the line (paragraph) of output. Then select the Word "Review" tab and click "Word Count" in the Proofing group on the far left.

- **JLetter.java**

**Requirements**: Write the application JLetter that displays letter J as a large block letter composed of the characters 'J', 'A', 'V', 'A' arranged in a pattern with a width of 12 characters and a height of 10 lines. Each line should begin in column 1 with appropriate leading spaces in lines 3, 4, 5, and 6. Trailing spaces at the end of each line should be avoided.

**Design**: Your program should contain a main method that prints exact the same pattern as shown in Figure 1. You should have 10 lines of output with each line in the pattern indicated.

```
JAVAJAVAJAVA
JAVAJAVAJAVA
       JAVA
       JAVA
       JAVA
       JAVA
J      JAVA
JA     JAVA
 JAVAJAVA
  JAVAJA
```

**Figure 1. Output from JLetter program**

**Code and Test**: The expected output contains 10 lines of letters in the specified order. Note that the first line of output has no leading spaces. Make sure that you print the pattern exactly as it appears above.

## Grading

**Web-CAT Submission**: Prior to submitting to Web-CAT, you should complete both programs and make sure that they appear to be working correctly. You will have 10 attempts to receive full credit for your two program files, but you should try to get everything right on the first submission. **You must submit both programs files at the same time. If you only submit one of the files, the submission will receive zero points for correctness**. Creating a jGRASP Project and adding the two files will make Web-CAT submission more straightforward. [Instructions for creating a jGRASP project can be found in Activity 1 (p. 5-6).]