UNIT 4 — TOUCHSTONE 4: Final Project

-/100

Not Submitted | Submitted | Graded

You can work on a Touchstone whenever you want, but you must complete the previous assessments in the Unit before you can submit it.

SUBMIT TOUCHSTONE

Currently, it takes about 3-7 business days for a Touchstone to be graded.

Touchstones are projects that illustrate your comprehension of the course material, help you refine skills, and demonstrate application of knowledge. You can work on a Touchstone anytime, but you must pass this unit's assessments before you submit it. Once you've submitted a Touchstone, it will be graded and counted toward your final course score.

Touchstone: Final Java Project

ASSIGNMENT: For this Touchstone, you will learn how to effectively plan, design, develop, and test an original program of your choosing. This program is your choice and it can be as complex as you wish. The goal is to start with an idea for your program and then step through the process of most phases of the development lifecycle to turn your idea into an actual program. This includes planning out the algorithm using pseudocode, coding your program using everything you learned from Units 1-3, and finally, testing and debugging your program to make sure it fulfills your intended purpose. You will fill out a journal (template link below) which has five sections that correspond to the five steps you will complete for your final project. Use this template to write your journal responses for each section of the project. When you have finished, you will submit this journal as a Word .doc/docx.

Java Journal Template

The following Example Java Journal Submission will be helpful to you as you work on this assignment and shows acceptable submissions for each entry.

Example Java Journal Submission

Being able to create your own program can be beneficial in any career field. Anything that can be programmed or automated in any industry could make your work a lot easier. Even as a hobby, it can give you a reference point to have a better understanding of how to deal with common issues. Using basic Java to automate menial tasks can save you time and money! Whether you were in this class to learn programming at a professional level or if it's just for fun, you will benefit from learning coding as a foundation of your technical skills.

In order to foster learning and growth, all work you submit must be newly written specifically for this course. Any plagiarized or recycled work will result in a Plagiarism Detected alert. Review Touchstones: Academic Integrity Guidelines for more about plagiarism and the Plagiarism Detected alert.

A. Directions

Step 1: Download and review the Java Journal Template.

Step 2: Review Section C (Requirements) to get acquainted with the requirements for this Touchstone before you begin writing your journal entries.

Step 3: Follow the directions for each part of the journal template. Make sure to include in your response all the elements listed under the Requirements section. Prompts in the Inspiration section are not required; however, they may help you to fully think through your response.

- Follow the journal creation process that is presented in your Unit 4 lessons to ensure you write a thorough journal entry. There is also an example of a successful journal entry for each submission in Section D (Additional Resources).
- Review Section B (Rubric) to make sure you have an understanding of how each journal entry will be graded.

Step 4: Once your journal is complete, including the code link, submit your journal for grading.

- There is a help video on "How to submit your Touchstone" in the first lesson of Unit 4 (Java Touchstone Overview).

B. Rubric

	Advanced (100%)	Proficient (85%)	Acceptable (75%)	Needs Improveme (50%)
PART 1: Defining Your Problem (16 points) State the problem you are planning to solve.	The student clearly stated the problem they are trying to solve, including what the program is expected to do to solve the problem and any necessary inputs and outputs	The student clearly stated the problem they are trying to solve, including answers to most of the required questions.	The student clearly stated the problem they are trying to solve, including answers to some of the required questions.	The student did r clearly state the problem they are to solve and/or d include answers most of the requi questions.

	Advanced (100%)	Proficient (85%)	Acceptable (75%)	Needs Improveme (50%)
	required for the program.			
PART 2: Working Through Specific Examples (16 points) Write down clear and specific steps to solve a simple version of the problem you identified in Part 1.	The steps to solve the problem are in English and easily understood and appear in logical order. The journal entry fully satisfies the requirements.	The steps to solve the problem are in English and easily understood and appear in logical order; however, some minor steps may be missing. The journal entry mostly satisfies the requirements.	The steps to solve the problem are in English but may not be in logical order or may be missing some steps. The journal entry barely satisfies the requirements.	The steps to solv problem are in E but highly illogica confusing, or mis critical step(s), ar the journal entry not satisfy the requirements.
PART 3: Generalizing Into Pseudocode (16 points) Write out the general sequence your program will use in pseudocode, including all specific examples you provided in Part 2.	The pseudocode submitted encapsulates the full functionality of the program and includes common program elements. The pseudocode is clearly laid out and readable to a person that does not know Java.	The pseudocode submitted encapsulates most of the functionality of the program, including common program elements. The pseudocode is clearly laid out and readable to a person that does not know Java.	The pseudocode submitted encapsulates most of the functionality of the program, including common program elements. However, the pseudocode is not clearly laid out and readable to a person that does not know Java.	The pseudocode submitted does n encapsulate the functionality of th program to inclu common prograr elements. The pseudocode is a clearly laid out ar readable to a per that does not kn Java.
PART 4: Testing Your Program (16 points) While writing and testing your program code, describe your tests, record any errors, and state your approach to fixing the errors.	Entry consists of a thorough explanation of the testing utilized and why, recording the error(s) found and the approach to fixing the error(s) for each of the test cases.	Entry consists of a thorough explanation of the testing utilized, including answering most of the required questions.	Entry consists of a thorough explanation of the testing utilized, including answering some of the required questions.	Entry does not gi thorough explana of the testing util and/or does not i answers to the re questions.
PART 5: Commenting Your Program (16 points) Submit your full program code, including thorough comments describing what each portion of the program should do when working correctly.	The provided code in the journal entry includes comments that thoroughly describe the program flow to a degree that a non-programming individual can easily follow the steps and conclude the outcome of the program.	The provided code in the journal entry includes comments that describe the program flow to a degree that a non-programming individual should be able to follow the steps and conclude the outcome of the program. There could be some additional comments needed or added clarity.	The provided code in the journal entry includes some comments on the program flow that may not be clear to a non-programming individual and/or may be missing some comments that would make the flow easier to understand.	The provided co the journal entry includes minimal comments to exp the program flow and/or is missing comments that w make the progra for a non-progra individual to be a follow the steps a conclude the out of the program.
PART 6: Your Completed Program (20 points) Provide the Replit link to your full program code.	The program works as expected. No errors present. Any required user input executed correctly. Output was as expected.	Performance level not used.	Performance level not used.	Program did not as expected; hov detailed commer were given as to the student felt it not run, including testing efforts.

C. Requirements

The following requirements must be met for your submission:

- Use a readable 11- or 12-point font.
- All writing must be appropriate for an academic context. Follow academic writing conventions (correct grammar, spelling, punctuation, and formatting).
- Journal entries must be original and written for this assignment; plagiarism of any kind is strictly prohibited.
- Submission must include your name and the date (included in the template).
- Include all of the journal parts in a single file. Acceptable file formats include .doc and .docx.
- Include your Replit share link at the appropriate location in the journal template.