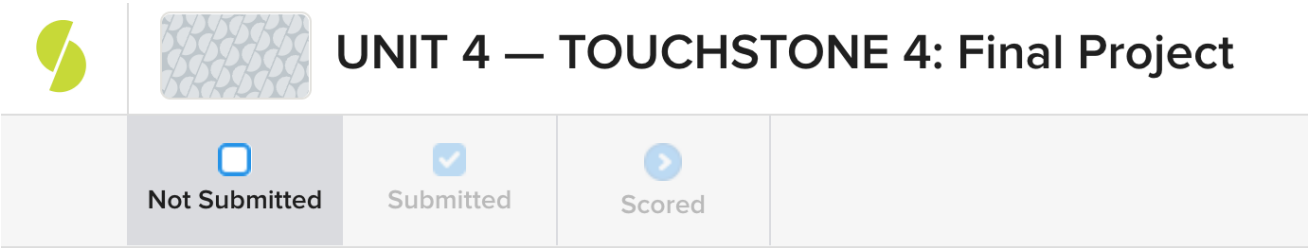
Java Touchstone Overview

by Sophia

WHAT'S COVERED

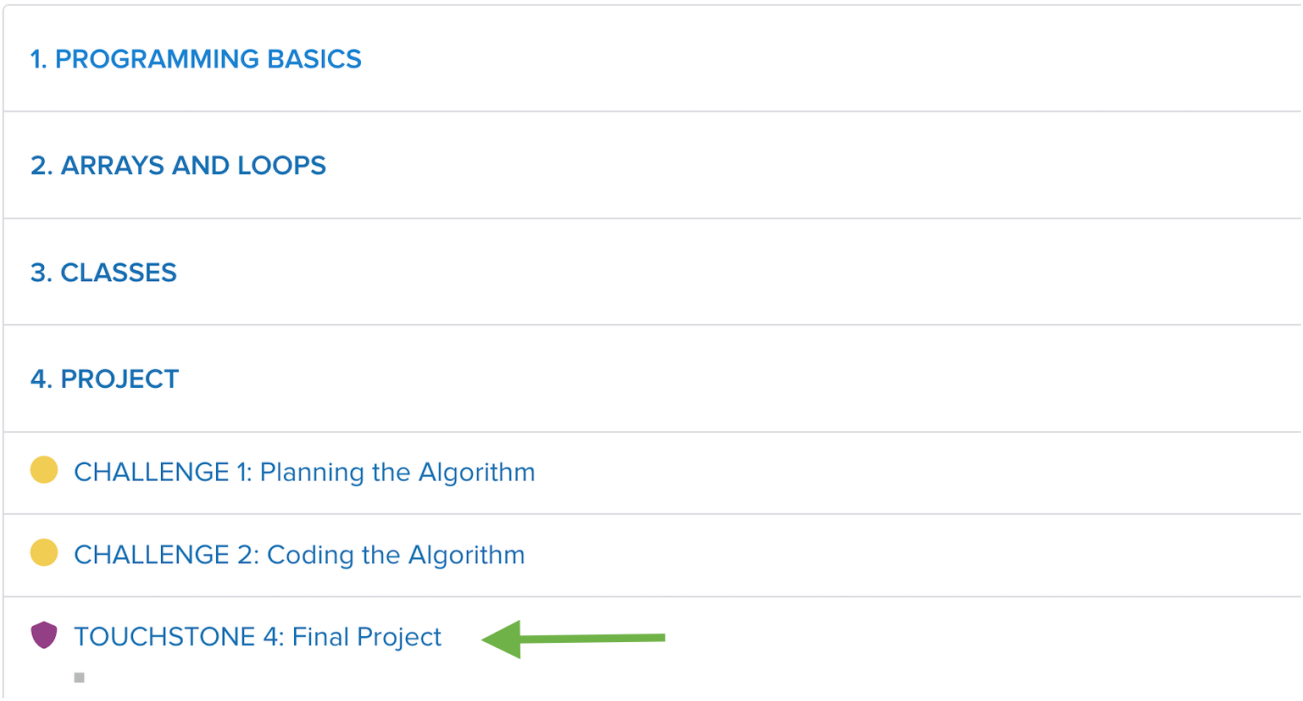
In this lesson, you will learn about all aspects of the final Touchstone, including requirements, grading, helpful aids, and submission details. Specifically, this lesson covers:

1. What Is the Touchstone?



The Touchstone is the pinnacle of this course. Up to this point, we have discussed Java syntax, and you have coded many programs. Now is the time to shine with what you've learned. The Touchstone is a human-graded activity that allows you to submit an actual program that you design, build, and test. Don't worry, we will be with you every step of the way. There are documents to help you and the lessons in Unit 4 will use an example program that can inspire your own program. Using what you have learned over the entirety of this course, it is time to design and code your own program!

Units



The Touchstone consists of one (1) submission. You will be submitting a Java Journal at the conclusion of Unit 4. However, this journal has six (6) entries. The easiest way to become familiar with this Java Journal is to visit the actual Touchstone page. The Java Journal's information is included as the last item in Unit 4 (see image above).

BIG IDEA

On the TOUCHSTONE 4: Java Final Project's page, you will see the following:

- Assignment directions with a link to the Java Journal Template
- Rubrics which will be used for grading the project
- Overall project requirements
- An additional resource that includes an Example Java Journal Submission document

TRY IT

Directions: Now that you know more about the Touchstone it is time to visit the Touchstone page and perform the following tasks:

1. Visit the Unit 4. PROJECT - TOUCHSTONE 4: Final Project page.
2. Carefully read through this page to get a better understanding of the overall project, the assignment directions, how grading works, the requirements for the project, and an additional resource. We suggest that you keep this page open while you follow the lesson, so you can easily reference each part.
3. Download the Touchstone JAVA Journal Template document and be prepared to open it on your device later in this lesson.

REFLECT

You may be thinking, "Touchstone, templates and journals, oh my!" These are some more new concepts to figure out. The good thing is, though, you are ready. You've learned not only a lot about Java throughout this

course, but also gained knowledge on some great coding concepts and methodologies that are applicable to any type of problem solving. The Touchstone work you are about to enter is just a way to capture what you now already know how to do—code your own program—and the templates and journals will document this journey. As you might hear before a classic boxing match—let's get ready to rumble, or rather, code!

---

2. What Is My Program?

THINK ABOUT IT

So, what program am I expected to design and code?

That is something you will think about as you move through Unit 4. Throughout this unit, you can follow along as you design and code a demonstration program. You will also create journal entries for this demo program throughout the unit. The lessons are designed to help you brainstorm your own program. There are inspirational questions for each part of the journal as well as the “Guided Brainstorming” section in each Unit 4 lesson to give you some additional thoughts.

Creating a program can seem daunting, but as you take each step through Unit 4, you will be adding to your ideas and given some thought-provoking questions to help. Your idea and program can be as simple or complex as you wish; just make sure to keep your journal entries tied to the requirements for each part of the journal. This can be a program for yourself, a friend, a family member, or anyone! You will be using a casino game as an example and bringing back thoughts and ideas on the drink program that we created together back in Unit 1. So, you can design and code your program to do anything you want. Maybe it's a program to convert measurements or collect movie reviews, or maybe it's a receipt saver or a financial planner. Really, the sky's the limit. However, remember to make sure it is something you feel you can design and build because it is a graded project.

HINT

It may seem obvious, but it needs to be stated clearly: please do not submit your Touchstone project using the demonstration program.

This project should be your own idea and code. Try to start simple and add as you progress throughout the unit. The Touchstone should only be submitted once you feel good about all of the entries you added to your journal. Feel free to rewrite, recode, and reestablish a new problem you would like to solve. Your project is only graded when you submit it on the Touchstone page in the course.

---

3. The Java Journal Template Document

The Java Journal Template document can be found in the Touchstone page of the course. Visit the Unit 4. PROJECT - TOUCHSTONE 4: Final Project page. In the Assignment section, you will notice a link for the Touchstone Java Journal Template document. Download this document and open it. Please read the directions at the top, if you haven't already.

3a. What Is in the Java Journal Template Document?

The template provides you with space for each part (entry) of the project. Below the first page, you will see each entry task along with the requirements that need to be addressed for a good score. When you are entering your journal entries throughout Unit 4, make sure you review the requirements for the entry since they are aligned with the rubric. Below the requirements for each entry, there are inspirational questions to help you start your brainstorming. Depending on the program you choose to design and build, they may or may not influence your ultimate entry submission.

TRY IT

Directions: Open your Touchstone Java Journal Template document and skim the tasks, requirements, and inspirations for the six (6) journal entries.

3b. When Do I Add an Entry to My Java Journal Template Document?

As you progress through the lessons in Unit 4, you will see a demonstration program being designed and coded in steps that match the Java Journal Template document. Feel free to type in and build the demonstration program in Replit to gain additional practice and insights, but remember you need to add journal entries for your own originally designed and coded program. Throughout Unit 4, you will see the demonstration program being entered into the Example Java Journal Submission document as entries. As you are working with your project, you will be encouraged to add entries about your project to your Java Journal.

3c. What Is the Example Java Journal Submission Document?

When you visit the Additional Resources area on the Touchstone page, you will see a link to the Example Java Journal Submission document. This is essentially a properly filled out Java Journal. This is the journal that is being created in lessons throughout Unit 4 with the demonstration program. You will have the opportunity to see some good journal entries as well as some not so good entries as you progress through Unit 4.

TRY IT

Directions: If you haven't already, now is a good time to download or view the Example Java Journal Submission document to get an idea of a well-laid-out Touchstone submission. You will find this document

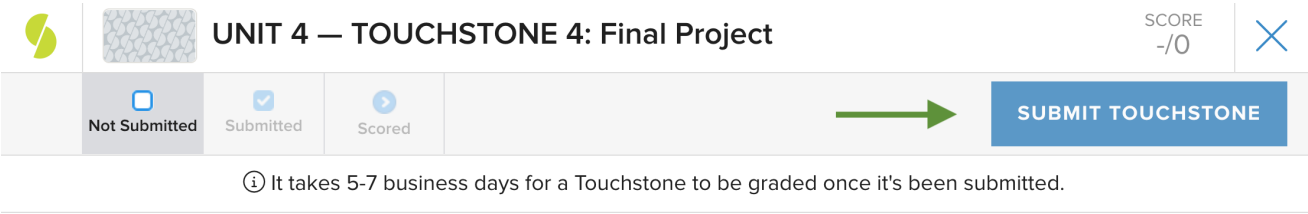
linked to the Unit 4 - Touchstone 4: Final Project page under Section D. Additional Resources.

3d. When Do I Submit My Java Journal?

You can only submit the Touchstone (your Java Journal) once, so make sure you have completed all the required entries (six in total) before submitting your Java Journal. Make sure all entries have good grammar and spelling in the earlier entries and that your code has been correctly pasted in the later entries. Make sure your Replit share link works and has been added to both the top page and as the last entry. Finally, ensure your name is added and the date you are submitting the Touchstone is on the first page.

3e. How Do I Submit My Java Journal?

Once you have your Java Journal Template filled out completely, you can submit the journal using the “Submit Touchstone” button at the top right of the Touchstone page.



HINT

Remember, you only have a one-time submission, so ensure your journal is complete before submitting. The final lesson also contains the directions to submit your journal.

---

4. Requirements and Grading

This Touchstone is worth 100 points. That breaks down to about 30% of the total course score, so make sure you take your time and review the requirements for the Java Journal. They are found on the Touchstone page in Section C. Requirements. You can see how your journal will be graded after submission by visiting Section B. Rubric. Each journal entry (Part 1 – 6) is a criteria row and each column is a level of performance that has an associated grade percentage. As you move from the left to the right, the percentage grade decreases. If the journal entry is missing any or all of the requirements listed on the Java Journal Template document, the grade received will decrease. Make sure you are familiar with the grading rubrics and review your journal entries before final submission.

---

5. Help Video

The following video demonstrates how to get a Replit share link that is both attached to the first page of the Java Journal Template document and also as the sixth journal entry.

WATCH

---

6. Unit 4 Overview

Each lesson of Unit 4 may consist of some review or mention of learning from a previous lesson; they are meant to prepare you for the Touchstone submission.

Here is a synopsis of each lesson in Unit 4. You will notice that with each lesson, we will continue with the completion of a demonstration program, and each will include a “Guided Brainstorming” section that will include ideas and extra examples to aid you in brainstorming your original idea that will eventually be coded into a program that you can submit for the Touchstone:

Challenge 1 (Planning the Algorithm)

1. **Java Touchstone Overview**. This lesson is the overview of the Touchstone, your program, Java Journal Template document, the Example Java Journal Submission document, and all requirements and grading of the Touchstone.
2. **Identifying a Problem to Solve**. This lesson will introduce you to the Unit demonstration program and walk through the questions you may need to ask to start the process of identifying a problem to solve. At the end of this lesson, we will demonstrate a good vs. bad example of a PART 1 entry of the Java Journal Template document. This would be a good time to add your journal entry as well.
3. **Working an Example**. In this lesson, we will continue to use the demonstration program to set up a series of steps required to solve the problem. Remember this is still early on in designing a program, but listing out the steps in a logical order is an important one. At the end of this lesson, we will again demonstrate a good vs. bad example for a PART 2 entry of the Java Journal Template document. This would be a good time to add your second journal entry as well.
4. **Identifying the Patterns**. In this lesson, we will be searching for patterns in the steps we listed out in the previous lesson. We will also be grouping steps and starting to look for any patterns that we can replicate.
5. **Forming the Algorithm**. In this lesson, we take the patterns and steps that we have identified and form algorithms based on them. Then, we can generalize our step-by-step algorithms into pseudocode. At the end of this lesson, we will demonstrate a good vs. bad example of a PART 3 entry of the Java Journal Template document. Once again, you will be reminded that while forming algorithms and pseudocode is fresh in your mind, this would be a good time to add your third journal entry.

Challenge 2 (Coding the Algorithm)

1. **Translating to Code**. In this lesson, we will take the demonstration program's pseudocode from the previous lesson's journal entry and start referencing code elements for the patterns.
2. **Writing the Program**. In this lesson, we will complete our coding of the demonstration program to include all the conditionals, loops, classes, etc., that we learned in this course.
3. **Testing As You Go**. In this lesson, we will see our demonstration program go through some tests that we learned about in past lessons. We will try to "break" the demonstration program. This is the QA (quality assurance) section, and since we do not have a dedicated QA individual/team, we need to do the test ourselves. At the end of this lesson, a good vs. bad example will be identified, and we will enter the PART 4 entry into the Java Journal Template document. As we have before, you will be reminded to add your fourth entry to your journal.
4. **Commenting Your Code**. In this lesson, we will see the demonstration program get the needed commenting so that the grader, or anyone for that matter, will be able to tell what our program is doing even if they are not Java-savvy. At the end of this lesson, a good vs. bad example regarding commenting will be shared, and we will finish the entry for PART 5 in the Java Journal Template document. This is also a chance to add comments to your program and enter it into your journal as the fifth entry.

5. **Course Wrap-Up**. In this lesson, we will obtain a Replit Share link and add that link as the final PART 6 entry into the Java Journal Template document.

SUMMARY

This lesson was set up to be an informational page for all things Touchstone-related. We discussed that this **Touchstone** is a human-graded project that involves the design and programming of a solution to an identified problem. We also discussed that this **program** can be as simple or complex as we feel comfortable with. The project submission will be based on the **Java Journal Template document**, which contains six journal entries to be written out during Unit 4 and submitted at the end. Also contained in the Java Journal Template document are directions, requirements, and inspirational questions to ask ourselves for **when we add each entry** of the journal. We reviewed how this project will be **graded** and some basic submission **requirements**. Finally, we identified some **support materials** including an **example submission** and reviewed what is expected to be covered and recommended tasks in each of Unit 4's lessons.

Source: This content and supplemental material has been adapted from Java, Java, Java: Object-Oriented Problem Solving. Source **cs.trincoll.edu/~ram/jjj/jjj-os-20170625.pdf**

It has also been adapted from "Python for Everybody" By Dr. Charles R. Severance. Source **py4e.com/html3/**