

Project 1 - Quiz

CSC-121, Fall 2025

Overview

- For your first project, you will create a program that incorporates user input into conditional logic. Specifically, you'll create a quiz or questionnaire on a topic of personal interest to you. The output a user receives from your program will differ based on how they respond to your questions.
- You can choose one of the following types of quizzes or questionnaires:
 - A knowledge-based quiz about something you care about (e.g. a music quiz, a sports quiz, an art quiz, a physics quiz)
 - A personality quiz that places users in a category based on their responses to fun questions (e.g. Which character are you from *Alice in Wonderland*?, Are you more like the planet Mercury, Mars, or Saturn?, Which 20th century decade should you be living in: the 1920s, the 1960s, or the 1980s?)
 - A questionnaire that leads to personalized recommendations (e.g. book recommendations, movie recommendations, game recommendations, vacation recommendations)

Project Requirements

- Create variables to store data that is required later in the program.
- In at least five places in the program, get input from the user and incorporate this input into your conditional logic.
- Include at least five conditional statements that consist of an `if` statement, and depending on the logic needed, `else if` and/or `else` statements. When appropriate, `switch` statements may be used instead of `if` statements.
- Use appropriate relational and logical operators in your `if` and `else if` statements.
- Display clear messages for users when soliciting their input and when sharing their results.

- Cleanly organize your code using proper Java code style (Following Java's class and variable naming conventions, proper indentation, etc.) so that it is easy to read and maintain.

Your Tasks

Part 1: Choose your focus

Your first step is to choose a focus for your project. Brainstorm by writing down several ideas that come to mind while thinking about these questions:

- What topics could you focus on? What are you interested in? What do you know a lot about? Engaging with a program that personally resonates with you can have positive effects on your learning experience.
- Which type of program would you be most interested in developing? Do you want to make a fact-based quiz or share recommendations? Do you want to come up with a creative personality quiz?
- What program would you enjoy sharing with your friends, family, or classmates? Thinking about other people interacting with your program might help you envision the user and what you'd like to share.

Then, narrow your ideas. First, keep your scope in mind. You want your chosen focus to make it possible for you to meet Project Requirements, such as getting input from users at least five times and incorporating this input into conditionals. And conversely, although you are encouraged to go a bit beyond minimum requirements, you also do not want your focus to be so large that you cannot complete your program within the project timeframe.

Further, you do not want your program to be too complicated, or require an understanding of programming concepts that we have not yet covered. Think about this question:

Which of your ideas work best with the scope of the project?

If multiple ideas work with the scope, imagine developing each idea in a little more detail. As you do that, you will get a greater sense of which project idea excites you the most. Finally, choose one idea and create a brief description of your chosen program. The description should include the focus of the program and details on what the final output to the user will be.

Example program descriptions

"I'm going to create a program that quizzes users about Impressionist art. User responses to questions will be used to award users points. The last question will give users who haven't scored any points a second chance to answer the question. At the end of the program, based on their points, users will get results about their level of knowledge on Impressionist art."

or

"My program will be a personality quiz that tells users if they would be happiest living in the 1920s, 1960s, or 1980s. I'll ask users a series of questions, and based on their answers, assign points to each decade. After they answer all the questions, they'll get a message about whether they are most suited for life in the 1920s, 1960s, or 1980s, followed by a brief description of that decade. If two decades tie as the most suitable, they'll get messages for both."

Part 2: Design your user prompts and potential responses

Next, make a list of the questions you'll ask users and their possible responses.

It's important that the questions have closed-ended rather than open-ended responses so that you can easily incorporate their answers into your conditional logic.

For instance, the following prompt for user input is open-ended:

None

Enter an Impressionist painter:

In response to this prompt, users could input the names of different painters (Cassatt, Renoir, Degas, etc.), or input variations of the same name (Claude Monet, monet, Monet, etc.). This creates challenges for your conditional logic because you won't be able to control all potential answers.

Instead, a closed-ended question would be:

None

Which of the following painters is an Impressionist?

1 - Monet

2 - Warhol

3 - Rembrandt

Enter 1, 2, or 3.

Users are given only three options to enter and each of these answers can be handled by the conditional logic. (For instance, an `if` statement can contain actions for a correct answer of 1 and an `else if` or `else` statement can contain actions for an incorrect answer of 2 or 3.)

Note: Even with this provided direction, there is still a chance users could enter something else like 4. Later, we'll learn programming concepts that will allow you to repeat the same question until user input conforms to expected responses.

For now, we'll assume that if closed options are given, users will input one of them. (Or, if you prefer, you can handle responses that do not align with given options by omitting the question from having an effect on final results and display a message about that.)

Part 3: Identify variables

A good next step as you plan is to think about how you'll keep track of important data in your program. Make a list of all the variables that you'll use in your program. As you do, think about these questions:

- What variables will you use to keep track of quiz or questionnaire results? Do you need a single variable to keep track of points or a level? Or, do you need multiple variables to represent potential results?
- What variables will you use to store user input?

Part 4: Determine the flow of your conditional logic

As a final part of your planning process, think about the flow of your conditional logic:

- Where will your conditional statements appear in relation to each other?
- What will be the components of each conditional statement? Which conditional statements will consist of just an `if` statement (if any)? Which will contain `else if` or `else` statements as well? Will you use switch statements?
- What will be the conditions in your `if` and `else` statements? Will any depend on more than one condition?
- What actions will occur when the body of each `if`, `else if`, and `else` statement executes? Will you increment or reassign variables? Will you print messages?
- Does your logic require any nested conditionals? This is not required, but is an option if certain conditional statements are only relevant when another condition is met. For instance, perhaps a user only answers a very difficult quiz question if they get an earlier question right.

Create an outline that identifies all of the conditional statements in the program, including the `if`, `else if`, `else`, and `switch` statements that comprise each conditional statement and the actions that occur when an `if`, `else if`, `else`, and `switch` statement executes. You can describe these actions using words rather than code.

Example outline for Impressionist quiz

- **Conditional statement 1:** Related to question 1 ("Are Impressionists known for creating detailed realistic scenes. Enter yes or no.")
 - `if` statement: when user response to question is no,
 - Displays congratulatory message
 - Increments value of level variable by 1
 - `else` statement: in all other cases,
 - Displays message that the answer is not correct and explains the correct answer

- **Conditional statement 2:** Related to question 2 ("Did Impressionist painters often focus on landscapes as their subject matter? Enter yes or no.")
 - if statement: when user response to question is yes,
 - Displays congratulatory message
 - Increments value of level variable by 1
 - else statement: in all other cases,
 - Indicates that the answer is not correct and explains the correct answer
- **Conditional statement 3:** Related to question 3 ("Which of the following painters is an Impressionist? 1 - Monet 2 - Warhol 3 - Rembrandt")
 - if statement: when user response to question is 1 for Monet,
 - Displays congratulatory message
 - Increments value of level variable by 1
 - else if statement: when user's level is 0 (meaning they need extra help) and response is 2,
 - Displays message that they are wrong but get one more chance and gives option of only 1 - Monet or 3 - Rembrandt
 - Executes nested conditional
 - **Conditional statement 4** (nested in conditional statement 3)
 - if statement: when user response is 1 for Monet,
 - Displays congratulatory message
 - Increments value of level variable by 1
 - else statement: in all other cases,
 - Indicates that the answer is not correct and explains the correct answer

- else if statement: when user's level is 0 (meaning they need extra help) and response is 3,
 - Displays message that they are wrong but get one more chance and gives option of only 1 - Monet or 2 - Warhol
 - Executes nested conditional 2
 - **Conditional statement 5** (nested in conditional statement 3)
 - if statement: when user response is 1 for Monet,
 - Displays congratulatory message
 - Increments value of level variable by 1
 - else statement:
 - in all other cases, Indicates that the answer is not correct and explains the correct answer
 - else statement: in all other cases,
 - Indicates that the answer is not correct and explains the correct answer
 - **Conditional statement 6:** Reveals results based on the value of the level variable
 - if statement: when level is 3,
 - Prints a congratulatory message that they answered all of the questions correctly and have a high level of knowledge about Impressionist painters
 - else if statement: when level is 1 or 2,
 - Prints a message that users have some knowledge about Impressionist painters and that encourages them to learn more
 - else if statement: when level is 0,
 - Prints a message that users didn't get any answers correct but that they can learn more

Part 5: Write your code

Next, refer to the planning you did in earlier parts and write the code that creates your program!

To make sure your program works as intended, compile and run the program multiple times using different user input each time.

Notes on genAI usage:

You're encouraged to use an AI tool while writing your code. For example, you may want to ask it to explain a concept related to your program, such as what operator to use to check if two values are equal. Or, if you find you have a syntax error, you could leverage genAI as part of your debugging process.

GenAI prompting suggestions

If you get stuck and want to try asking an AI tool for help, here's some example prompts you might want to try.

- In Java, if I want to perform a calculation with user input, what data type must that user input be in?
- How can I perform string concatenation and calculations within a single `System.out.println()` statement in Java?
- How is `System.out.print(...)` different from `System.out.println(...)`?
- What will happen if I don't have either a `System.out.print(...)` different from `System.out.println(...)` before a call to `scanner.nextInt()`
- How is `scanner.nextInt()` different from `scanner.nextLine()`?

Part 6: Explain your work

A final part of the project is to explain your work and demonstrate your understanding of the program you created.

You will share your explanation in a 5-minute video after creating a detailed outline of the points that you want to make.

Specifically, you must explain the following:

- The intended functionality of the program you decided to create and why you chose to create that program
- Your process and the steps you took to design and develop this program
- How the code executes in one of your most complex conditional statements

- The lessons you learned through this project

Each part should be at least 1-minute long. You can use a standard cellphone or webcam to record your video.

Deliverables and Evaluation Criteria

You will be responsible for the following deliverables. The weighting of each component is indicated below. It's important that you refer to the Project 1 Rubric for detailed information about how each deliverable will be evaluated.

Checkpoint (10%)

The following deliverables must be shared with a TA in an in-person meeting by the checkpoint date:

- A brief description of the program's focus and the final results users will get
- A list of the questions that users will be prompted with and possible responses
- A list of variables with their Java types that will be used in the program
- An outline of the conditional statements in the program, including the `if`, `else if`, `else`, and `switch` statements that comprise each conditional statement and the actions that occur when an `if`, `else if`, `else`, or `switch` statement executes

Code submission (60%)

You must submit your code by the due date. *Please ensure it's possible to compile and successfully run your Java program!*

Explanation (30%)

You must submit your video containing your explanation along with your detailed outline by the due date.