






KIT101 Programming Fundamentals

PP Task 4.1 If This Then That

Overview

- Purpose:** Learn how to use conditional control flow statements in a program.
- Task:** Use the instructions on the following pages to complete the implementation of a coffee order generator, and trace some small examples of conditional control flow. Submit to MyLO when complete.
- Learning Outcomes:** 1  2 
- Time:** Aim to complete this task before the start of Week 5.
- Resources:**
- Introductory Programming Notes:
 - 08 Making Decisions
 - Online Lecture Materials & Echo360:
 - Making Decisions:
 -  Boolean expressions & comparing values
 -  Two-way branching with if and if-else
 -  Multi-way branching with switch

Submission Details

Upload the following to the MyLO submission folder for this task:

- **Tracing tables** for the code samples at the start. This can be *either* photos/scans of hand-written tables (recommended) or a Word/PDF document
- Your **CoffeeOrder.java** source file
- A **screenshot** showing the execution of your CoffeeOrder program

Assessment Criteria

A  Completed submission will:

- Include valid tracing tables for the [code samples](#) in the warm up exercise, showing the execution order and output produced

- Demonstrate the use of if, if-else or switch to perform selection, as appropriate to the task
- Produce the correct output for any valid inputs
- Follow the unit's coding style for layout, including indentation within selection statements
- Compile and run, with the screenshot showing that it works

Instructions

Warm up exercise: Trace code involving selection

1. Create tracing tables for the following code samples. The tracing tables should illustrate the execution path each program took, both by the sequence of row numbers in the table and the output messages produced. Remember to include your name and the name of the task in the document(s) you submit.

Selection Sample 1:

```
1  int age = 5;
2  System.out.println("Message 0");
3  if (age < 6) {
4      System.out.println("Message 1");
5  } else {
6      System.out.println("Message 2");
7  }
8  System.out.println("Message 3");
```

Selection Sample 2:

```

1  int age = 7;
2  System.out.println("Message 0");
3  if (age < 6) {
4      System.out.println("Message 1");
5  } else {
6      System.out.println("Message 2");
7  }
8  System.out.println("Message 3");

```

Implement a program that uses selection

In this task you will complete the implementation of a coffee order generator, which asks the user a series of questions about their order and then presents a concise formatted summary.

- Download and extract the **4.1PP IFTTT Starter Code** from MyLO.
- Open **CoffeeOrder.java** in DrJava and read through the code. The program reads a user's coffee preferences—kind of coffee, number of shots of coffee, and whether they want soy milk instead of regular milk—and then generates a summary of their order. Or rather, it *will* generate that summary when you have finished.

Note: Before you complain that this is unrealistic, yes we know that it doesn't really allow for coffees without milk. This is to make the problem easier to solve.

The rules for the output message are as follows:

- The complete message must appear on a single line, ending with a newline (but can be generated piece by piece).
- If the number of shots is one, output 'Single shot'. If the number of shots is two, output 'Double shot'. Otherwise output 'Dark and bitter'. (You can assume they will only enter values ≥ 1 .)
- If the user has requested soy milk then output 'soy'.
- Conclude the output message with their coffee name (the text they entered earlier)

- Compile and test your program. You can use the following sample input values and expected outputs to test that your program works:

coffee	shots	useSoy	Expected output
latte	1	false	Order details: Single shot latte

coffee	shots	useSoy	Expected output
cappuccino	2	true	Order details: Double shot soy cappuccino
macchiato	3	false	Order details: Dark and bitter macchiato
espresso	1	true	Order details: Single shot soy espresso

Tip: Test your program using *at least* the examples in the table above, then try a few alternatives of your own devising. Remember that its output must match what's expected, including being all on one line.

5. **Take a screenshot** of one run through your program.
 6. Ensure you've completed the [code tracing exercises](#) from the start and then upload the code traces, completed CoffeeOrder.java source file and screenshot and to the submission folder for this task.
-