System Requirements Specification Index

For

If Else If Statement

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Table of Contents

USE CASE DESCRIPTION
System Requirements Specification

1 Project Abstract

This project assesses knowledge of Java conditional statements, specifically the **if-else-if** statement.
The tasks involve checking multiple conditions and making decisions based on numerical values, logical comparisons, and complex decision-making scenarios.

2 Assessment Tasks

Task 1: Check if a Number is Positive, Negative, or Zero:

- Declare an integer variable `number` with an initial value of `0`.
- Use an `if-else-if` structure to check the following conditions:
  - ➔ Condition 1:
    - ➢ Use `if (number > 0)` to check if `number` is greater than `0`.
    - ➢ If true, print `"The number <number> is positive."`.
  - ➔ Condition 2:
    - ➢ Use `else if (number < 0)` to check if `number` is less than `0`.
    - ➢ If true, print `"The number <number> is negative."`.
  - ➔ Else Condition:
    - ➢ Use `else` to handle the case when `number` is neither positive nor negative (i.e., it is `0`).
    - ➢ Print `"The number <number> is zero."`.

Task 2: Find the Smallest of Three Numbers:

- Declare and initialize three integer variables:
  - ➔ `a` with the value `10`.
  - ➔ `b` with the value `5`.
  - ➔ `c` with the value `15`.
- Use an `if-else-if` structure to compare the values of `a`, `b`, and `c` as follows:
  - ➔ Condition 1:
    - ➢ Use `if (a <= b && a <= c)` to check if `a` is less than or equal to both `b` and `c`.
    - ➢ If true, print `"The smallest number is: <a>"`.
  - ➔ Condition 2:
    - ➢ Use `else if (b <= a && b <= c)` to check if `b` is less than or equal to both `a` and `c`.
    - ➢ If true, print `"The smallest number is: <b>"`.

- ➔ Else Condition:
  - ➢ Use `else` to handle the case when both `a` and `b` are greater than `c`.
  - ➢ Print `"The smallest number is: <c>"`.

Task 3: Check if a Number is Divisible by 3, 5, or Both:

- Declare an integer variable `num` with an initial value of `15`.
- Use an `if-else-if` structure to check the following conditions:
  - ➔ Condition 1:
    - ➢ Use `if (num % 3 == 0 && num % 5 == 0)` to check if `num` is divisible by both `3` and `5`.
    - ➢ If true, print `"The number <num> is divisible by both 3 and 5."`.
  - ➔ Condition 2:
    - ➢ Use `else if (num % 3 == 0)` to check if `num` is divisible by `3` only.
    - ➢ If true, print `"The number <num> is divisible by 3."`.
  - ➔ Condition 3:
    - ➢ Use `else if (num % 5 == 0)` to check if `num` is divisible by `5` only.
    - ➢ If true, print `"The number <num> is divisible by 5."`.
  - ➔ Else Condition:
    - ➢ Use `else` to handle the case when `num` is not divisible by either `3` or `5`.
    - ➢ Print `"The number <num> is divisible by neither 3 nor 5."`.

Task 4: Grade Calculation Based on Marks:

- Declare an integer variable `marks` with an initial value of `82`.
- Use an `if-else-if` structure to calculate the grade based on the following conditions:
  - ➔ Condition 1:
    - ➢ Use `if (marks >= 90)` to check if `marks` is greater than or equal to `90`.
    - ➢ If true, print `"Grade: A"`.
  - ➔ Condition 2:
    - ➢ Use `else if (marks >= 75)` to check if `marks` is between `75` and `89`.
    - ➢ If true, print `"Grade: B"`.
  - ➔ Condition 3:
    - ➢ Use `else if (marks >= 50)` to check if `marks` is between `50` and `74`.

- ➢ If true, print `"Grade: C"`.

➔ Else Condition:

- ➢ Use `else` to handle the case when `marks` is less than `50`.
- ➢ Print `"Grade: F"`.

Task 5: Check Leap Year:

- Declare an integer variable `year` with an initial value of `2024`.
- Use an `if-else` structure to check if the year is a leap year:

➔ Condition:

- ➢ Use `if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))` to check if `year` is a leap year.
- ➢ If true, print `"The year <year> is a leap year."`.

➔ Else Condition:

- ➢ Use `else` to handle the case when the `year` is not a leap year.
- ➢ Print `"The year <year> is not a leap year."`.

Expected Output:

The number 0 is zero.
The smallest number is: 5
The number 15 is divisible by both 3 and 5.
Grade: B
The year 2024 is a leap year.

3 Template Code Structure

3.1 Package: com.yaksha.assignment.IfElseIfStatementAssignment

Resources

Class/Interface	Description	Status
IfElseIfStatementAssignment (class)	• Main class demonstrating conditional checks using `if-else-if` statements. • Includes examples of: - Checking positive, negative, or zero using `if-else-if`. - Finding the smallest of three numbers using `if-else-if`.	Need to be implemented.

	- Checking divisibility by 3, 5, or both using `if-else-if`. - Calculating grades using `if-else-if` conditions. - Checking leap year using `if-else-if` logic.	

4 Execution Steps to Follow

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) → Terminal →New Terminal.**
3. **This editor Auto Saves the code.**
4. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**
5. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
6. **To run your project use command:**
   mvn compile exec:java
   -Dexec.mainClass="com.yaksha.assignment.IfElseIfStatementAssignment"
7. **To test your project test cases, use the command**
   mvn test
8. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**