System Requirements Specification Index

For

Switch Case Operator

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 control flow statements, specifically the **switch-case** statement.

The tasks involve evaluating expressions and making decisions based on constant values using switch-case constructs, covering scenarios like days, months, grades, and seasons.

2 Assessment Tasks

Task 1: Day of the Week Based on the Given Integer Using Switch-Case:

- Declare an integer variable `day` with an initial value of `3`.
- Declare an empty `String` variable `dayOfWeek` to store the day name.
- Use a `switch` statement on `day` and check the following cases:
  - ➔ **Case 1:** Assign `"Monday"` to `dayOfWeek`.
  - ➔ **Case 2:** Assign `"Tuesday"` to `dayOfWeek`.
  - ➔ **Case 3:** Assign `"Wednesday"` to `dayOfWeek`.
  - ➔ **Case 4:** Assign `"Thursday"` to `dayOfWeek`.
  - ➔ **Case 5:** Assign `"Friday"` to `dayOfWeek`.
  - ➔ **Case 6:** Assign `"Saturday"` to `dayOfWeek`.
  - ➔ **Case 7:** Assign `"Sunday"` to `dayOfWeek`.
  - ➔ **Default Case:** Assign `"Invalid day number"` to `dayOfWeek`.
- Print the result using `System.out.println("Day of the week: " + dayOfWeek)`.

Task 2: Number of Days in a Month Based on the Month Number Using Switch-Case:

- Declare an integer variable `month` with an initial value of `2`.
- Declare an empty `String` variable `daysInMonth` to store the number of days.
- Use a `switch` statement on `month` and check the following cases:
  - ➔ **Cases 1, 3, 5, 7, 8, 10, 12:** Assign `"31 days"` to `daysInMonth`.
  - ➔ **Cases 4, 6, 9, 11:** Assign `"30 days"` to `daysInMonth`.
  - ➔ **Case 2:** Assign `"28 or 29 days"` to `daysInMonth`.
  - ➔ **Default Case:** Assign `"Invalid month number"` to `daysInMonth`.
- Print the result using `System.out.println("Number of days in month: " + daysInMonth)`.

Task 3: Grade based on marks using switch-case:

- Declare an integer variable `marks` with an initial value of `82`.

- Declare an empty `String` variable `grade` to store the grade.
- Use a `switch` statement on `marks / 10` and check the following cases:
  - ➔ **Cases 10, 9:** Assign `"Grade: A"` to `grade`.
  - ➔ **Case 8:** Assign `"Grade: B"` to `grade`.
  - ➔ **Case 7:** Assign `"Grade: C"` to `grade`.
  - ➔ **Default Case:** Assign `"Grade: F"` to `grade`.
- Print the result using `System.out.println(grade)`.

Task 4: Month Name Based on the Month Number Using Switch-Case:

- Reassign `month` with the value `5`.
- Declare an empty `String` variable `monthName` to store the name of the month.
- Use a `switch` statement on `month` and check the following cases:
  - ➔ **Case 1:** Assign `"January"` to `monthName`.
  - ➔ **Case 2:** Assign `"February"` to `monthName`.
  - ➔ **Case 3:** Assign `"March"` to `monthName`.
  - ➔ **Case 4:** Assign `"April"` to `monthName`.
  - ➔ **Case 5:** Assign `"May"` to `monthName`.
  - ➔ **Case 6:** Assign `"June"` to `monthName`.
  - ➔ **Case 7:** Assign `"July"` to `monthName`.
  - ➔ **Case 8:** Assign `"August"` to `monthName`.
  - ➔ **Case 9:** Assign `"September"` to `monthName`.
  - ➔ **Case 10:** Assign `"October"` to `monthName`.
  - ➔ **Case 11:** Assign `"November"` to `monthName`.
  - ➔ **Case 12:** Assign `"December"` to `monthName`.
  - ➔ **Default Case:** Assign `"Invalid month number"` to `monthName`.
- Print the result using `System.out.println("Month name: " + monthName)`.

Task 5: Season Based on the Month Number Using Switch-Case:

- Reassign `month` with the value `12`.
- Declare an empty `String` variable `season` to store the season name.
- Use a `switch` statement on `month` and check the following cases:
  - ➔ **Cases 12, 1, 2:** Assign `"Winter"` to `season`.
  - ➔ **Cases 3, 4, 5:** Assign `"Spring"` to `season`.
  - ➔ **Cases 6, 7, 8:** Assign `"Summer"` to `season`.
  - ➔ **Cases 9, 10, 11:** Assign `"Fall"` to `season`.
  - ➔ **Default Case:** Assign `"Invalid month number"` to `season`.
- Print the result using `System.out.println("Season: " + season)`.

Expected Output:

Day of the week: Wednesday
Number of days in month: 28 or 29 days
Grade: B
Month name: May
Season: Winter

3 Template Code Structure

3.1 Package: com.yaksha.assignment.SwitchCaseAssignment

Resources

Class/Interface	Description	Status
SwitchCaseAssignment(class)	• Main class demonstrating control flow using **switch-case** statements. • Includes examples of: - Mapping integer values to days of the week using **switch-case**. - Determining the number of days in a month based on the month number. - Grade calculation based on marks. - Determining month names. - Determining seasons based on the month number.	Need to be implemented.

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.SwitchCaseAssignment"
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.**