| Dr. Sami Albouq |
| Associate Professor |
| Email: salbouq1@iu.edu.sa |

Lab 3: Understanding flow of control

Objective: The objective of this lab is to provide students with a comprehensive understanding of various control structures in Java programming. By covering boolean expressions, relational operators, logical operators, as well as if statements, else statements, if-else if statements, and switch statements, students will gain practical experience in implementing these essential programming constructs. Through hands-on exercises and examples, students will learn how to manipulate program flow, make decisions, and respond to different scenarios, thereby building a strong foundation in control structures that are fundamental to programming in Java.

Part 1: Boolean Expressions and Relational Operators

1. Open your Java IDE and create a new Java class named BooleanOperatorsLab.
2. Define a main method within the class.
3. Inside the main method, declare two integer variables named num1 and num2.
4. Import the required library to facilitate user input.
5. Create a Scanner object to read input from the user.
6. Prompt the user to enter two integer values for num1 and num2.
7. Create a boolean variable named result1 and assign it the value of the expression (num1 > num2).
8. Display the value of result1 along with a descriptive message using System.out.println().
9. Create another boolean variable named result2 and assign it the value of the expression (num1 == num2).
10. Display the value of result2 along with an appropriate message.

Instructions:
- Begin by importing the necessary Scanner library.
- Utilize the Scanner class to read user input.
- Follow the instructions and provided descriptions for each step.

Expected Output (example input: num1 = 10, num2 = 5):

```
Enter the value for num1: 10
Enter the value for num2: 5
Result 1 (num1 > num2): true
Result 2 (num1 == num2): false
```

Part 2: Logical Operators

1. Continuing from the previous code, create a new boolean variable named result3.
2. Assign it the value of the expression (result1 && result2).
3. Print the value of result3 along with a message indicating it represents the result of a logical AND operation.
4. Create another boolean variable named result4.
5. Assign it the value of the expression (result1 || result2).
6. Print the value of result4 along with a message indicating it represents the result of a logical OR operation.

Instructions:
- For steps 2 and 5, use the logical AND (&&) and logical OR (||) operators respectively.
- Use the System.out.println() method to display the results.

Expected Output:
Result 3 (result1 && result2): false
Result 4 (result1 || result2): true
Part 3: Using if statement.

Task 1: Checking Eligibility for Discount

1. Open your Java IDE and create a new Java class named CheckEligibility.
2. Define a main method within the class.
2. Import the necessary library to enable user input.
3. Declare two integer variables: age and totalAmount.
4. Create a Scanner object to read input from the user.
5. Prompt the user to enter their age.
6. Prompt the user to enter the total amount of purchase.
7. Write an if statement with a boolean expression that checks whether the person is eligible for a discount. The conditions for eligibility are:
   - Age is between 18 and 30 (inclusive).
   - Total amount is greater than or equal to 1000.
8. If the conditions are met, display "You are eligible for a discount!" using the System.out.println() method.

Instructions:
- Import the necessary Scanner library.
- Utilize the Scanner class to read user input for age and totalAmount.
- Write an if statement to check the eligibility conditions.
- Use the System.out.println() method to display the result.

Expected Output (example input: age = 25, totalAmount = 1500):
Enter your age: 25
Enter the total amount of your purchase: 1500
You are eligible for a discount!

Task 2: Membership Renewal

1. Declare a boolean variable isMember.
2. Import the necessary library to enable user input.
3. Create a Scanner object to read input from the user.
4. Prompt the user to enter whether they are a member (true or false).
5. Declare an integer variable yearsOfMembership.
6. Prompt the user to enter the number of years they have been a member.
7. Write an if statement with a boolean expression that checks whether the person is a member and has been a member for at least 2 years.
8. If the conditions are met, display "You are eligible for membership renewal!" using the System.out.println() method.

Instructions:
- Import the necessary Scanner library.
- Utilize the Scanner class to read user input for isMember and yearsOfMembership.
- Write an if statement to check the eligibility conditions.
- Use the System.out.println() method to display the result.

Expected Output (example input: isMember = true, yearsOfMembership = 3):

Are you a member? (true/false): true
How many years have you been a member? 3
You are eligible for membership renewal!

Part 5 Using else statements.

Task 1: Adding else to Part 3 (Task 1 and 2)

1. Recall the scenario from Part 3 (Task 1) where we checked eligibility for a discount based on age and total amount.
2. after the if block that prints "You are eligible for a discount!", add an else statement.
3. inside the else block, print "Sorry, you are not eligible for a discount."

Task 2: Adding else to Part 3

1. Recall the scenario from Part 3 (Task 2) where we checked membership renewal eligibility based on membership status and years.
2. After the if block that prints "You are eligible for membership renewal!", add an else statement.
3. Inside the else block, print "You are not eligible for membership renewal."

Observations:

Observe the behavior of the if statements in different scenarios and analyze how the boolean expressions using relational and logical operators affect the flow of the program.

Part 6 Using if else if statement.

1. Open your Java IDE and create a new Java class named ConditionalStatementsLab.
2. Define a main method within the class.
3. Import the necessary library to facilitate user input.
4. Create a Scanner object to read input from the user.
5. Prompt the user to enter their age.
6. Implement an if-else-if structure to determine the stage of life based on the age:
   - If age is less than 0, display "Invalid age".
   - If age is less than 18, display "You are a minor".
   - If age is between 18 and 65, display "You are an adult".
   - If age is 65 or older, display "You are a senior citizen".

Expected Output (example input: age = 25):

```
Enter your age: 25
You are an adult.
```

Part 7 Using Switch statement.

1. Open your Java IDE and create a new Java class named IntegerSwitchLab.
2. Define a main method within the class.
3. Declare an integer variable named dayOfWeek and assign it a value between 1 and 7 to represent a day of the week (e.g., 3 for Wednesday).
4. Implement a switch statement to display the name of the day of the week based on the value of dayOfWeek:
   - Case 1: Print "Monday"
   - Case 2: Print "Tuesday"
   - Case 3: Print "Wednesday"
   - Case 4: Print "Thursday"
   - Case 5: Print "Friday"
   - Case 6: Print "Saturday"
   - Case 7: Print "Sunday"
   - Default case: Print "Invalid day"

Instructions:

- Use the switch, case, and default keywords.
- The cases should correspond to the integer values representing days of the week.
- Make sure to include a break statement at the end of each case.

Questions:

1. You're tasked with developing a Java program for a local bookstore to help customers determine their book purchase discount. The program should allow customers to input the number of books they plan to buy. Based on this input, the program will automatically calculate and display the applicable discount percentage. Depending on the quantity of books chosen, different discount rates will be applied.

   Example Output:
   Enter the number of books: 8
   You've chosen to buy 8 books.
   Congratulations! You qualify for a 10% discount on your purchase.

2. You are developing a Java program for a university's course enrollment system. The system needs to determine which courses a student is eligible to enroll in based on their academic standing and the prerequisites for each course. Implement a complex if-else if structure to handle various scenarios involving nested conditions.

   A student's eligibility depends on the following factors:

   - If the student's GPA is 3.5 or higher, they are eligible to enroll in any course.
   - If the student's GPA is between 3.0 and 3.49, they can enroll in any course with no prerequisites.
   - If the student's GPA is below 3.0, they can only enroll in courses with prerequisites they have fulfilled.

   Furthermore, courses have different prerequisites:

   - Course A requires a GPA of 3.0 or higher.
   - Course B requires a GPA of 3.5 or higher.
   - Course C requires a GPA of 2.5 or higher and completion of Course A.
   - Course D requires a GPA of 3.0 or higher and completion of Course B.

   Write a Java program that takes the student's GPA and a boolean value indicating whether the student has completed Course A as input. Based on this input, the program should determine and display the courses the student is eligible to enroll in.

   Example Output:
   Enter your GPA: 3.2
   Have you completed Course A? (true/false): true

   You are eligible to enroll in the following courses:
   - Course C
   - Course D

   Instructions:
   - Use the Scanner class to read user input.
   - Implement nested if and else if statements to handle the complex eligibility criteria.
   - Display the eligible courses based on the conditions met.

   Note:
   This question tests your ability to work with complex eligibility conditions involving both GPA and course prerequisites within nested if and else if structures.

Conclusion:
In this lab, students have delved into the world of control structures, essential tools for effective decision-making and program flow management in Java. They've explored boolean expressions and relational operators to evaluate conditions, logical operators to combine conditions, and worked with different flavors of if statements to execute code blocks conditionally. The incorporation of else statements and the if-else if statements has expanded their understanding of branching scenarios, allowing for multiple paths of execution. Lastly, students have discovered the utility of switch statements for handling various cases efficiently. This comprehensive exposure equips them with the skills to craft dynamic and responsive programs, enhancing their overall programming prowess in Java.