CodInClub

Powered By BridgeLabz

Programming Concept Practiced

1. **Java Control Flows -** All the programs done under Programming Construct ensured that every line of code was executed. However, in many situations, we may run some lines of code and skip others using Control Flows using Conditions and Loops
   a. Create programs that can make decisions using Conditional Statements
   b. execute code repeatedly using Loops
2. **Java *boolean* Data Type -** ***boolean*** is a data type that can have true or false
   a. The ***boolean*** values true and false are used without quotation marks. For example, ***true*** not ***"true"***
   b. ***true*** and ***false*** are not variables. We cannot create variables named true and false.
   c. They must be in lowercase. For example, ***true*** not True, and ***false*** not False.
3. **Boolean Expressions -** Boolean Expressions are the ones that use the comparison operator and the resultant of the expression is either ***true*** or ***false***. Here is the list of Comparison Operators

Operators	Expression	Remarks
`==`	`6 == 5`	checks if the value `6` is **equal to** `5`
`!=`	`6 != 5`	checks if the value `6` is **not equal to** `5`
`>`	`6 > 5`	checks if the value `6` is **greater than** `5`
`>=`	`6 >= 5`	checks if the value `6` is **greater than or equal to** `5`
`<`	`6 < 5`	checks if the value `6` is **less than** `5`
`<=`	`6 <= 5`	checks if the value `6` is **less than or equal to** `5`

```java
// Is the given number even or odd

// Take input for a number
System.out.print("\nEnter a number: ");
int number = input.nextInt();

// Display if the number is even or odd
System.out.println("Is the number Even? " + (number % 2 == 0));
System.out.println("Is the number Positive? " + (number > 0));
System.out.println("Is the number Zero? " + (number == 0));
```

4. **Logical Operators -** Logical Operators allows to combine multiple boolean expressions. Here is the list of Logical Operators

Operator	Meaning	Example
`&&` (AND)	`true` if both expressions are true	`(age > 17) && (salary > 4000)`
`||` (OR)	`true` if either of the expressions is true	`(age > 17) || (salary > 4000)`
`!` (NOT)	`true` if the expression is `false`	`!(age > 17)`

```java
// Take input for a number
System.out.print("\nEnter a number: ");
int number = input.nextInt();

// Display the number characteristics
System.out.println("Is number Even & greater or equal to 10 ? " +
                   (number % 2 == 0) && number >= 10);
System.out.println("Is the number divisible by 3 or 4? " +
                   (number % 3 == 0 || number % 4 == 0));
System.out.println("Is the number not zero and divisible by 3? " +
                   (!(number == 0) && number % 3 == 0));
```

5. **The *if* Statement -** The *if* Statement is a Conditional Statement that evaluates a Boolean Expression as ***true*** or ***false***. The syntax is of the form shown below.

```
if (number % 2 == 0) {
    System.out.println("The number is a Even Number”);
}
```

Note: The line of code inside if block is executed only if the boolean expression `if (number % 2 == 0)` is ***true***. If the boolean expression is ***false***, the line of code inside the block is skipped.

E.g. => Write a program to check a student's grade and if it is greater than or equal to 50 (grade >= 50 is true), the student passed the examination. In this case, "congratulations message" is executed. If the student's grade is less than 50 (age >= 50 is false), the student failed the examination. In this case, "try again" message is executed.

Java

```
import java.util.Scanner;

class StudentResultIndicator {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // get student score from the user
        System.out.println("Enter grade");
        int studentGrade = input.nextInt();

        // if studentGrade is 50 or above, studentGrade >= 50 results to true
        if (studentGrade >= 50) {
            System.out.println("Congratulations!");
            System.out.println("You passed the examination.");
        }

        // if student_grade is less than 50
        if (studentGrade < 50) {
            System.out.println("Sorry.");
            System.out.println("You failed the examination.");
        }
        input.close();
    }
}
```

6. **Java *else* Clause -** The ***if*** statement can have an optional ***else*** statement. The syntax is

```
if (number % 2 == 0) {
    System.out.println("The number is a Even Number”);
} else {
    System.out.println("The number is not a Even Number”);
}
```

E.g. => Lets Solve the Pass or Fail with ***if*** and ***else*** block

```
Java
import java.util.Scanner;

class StudentResultIndicator {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // get student score from the user
        System.out.println("Enter grade");
        int studentGrade = input.nextInt();

        // if studentGrade is 50 or above
        if (studentGrade >= 50) {
            System.out.println("Congratulations!");
            System.out.println("You passed the examination.");
        } else {
            // Student_grade is less than 50
            System.out.println("Sorry.\nYou failed the examination.");
        }
        input.close();
    }
}
```

7. **Java *else if* Clause -** As we have seen, the if statement with the else clause allows us to choose between two different options. If we need to choose among more than two options, we can add the else if clause. The syntax is

```
if (number % 2 == 0) {
    System.out.println("The number is a Even Number”);
} else if (number % 2 == 1) {
    System.out.println("The number is a Odd Number”);
} else {
   System.out.println("The number is not a Natural Number”);
}
```

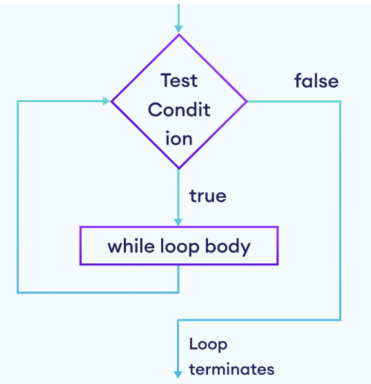
E.g. => Let's perform check for Invalid Grades and then Solve the Pass or Fail

```
Java
import java.util.Scanner;

class StudentResultIndicator {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // get student score from the user
        System.out.println("Enter grade");
        int studentGrade = input.nextInt();

        // condition for invalid score
        if (studentGrade < 0 || studentGrade > 100) {
            System.out.println("Invalid grade.");
        }
        // if is changed to else if
        else if (studentGrade >= 50) {
            // if studentGrade is 50 or above
            System.out.println("Congratulations!");
            System.out.println("You passed the examination.");
        } else {
            // Student_grade is less than 50
            System.out.println("Sorry.\nYou failed the examination.");
        }
        input.close();
    }
}
```

8. **Java *while* loop -** The ***while*** loop evaluates the boolean expression. If it's ***true***, the body of the while loop is executed.
    a. It's similar to the ***if*** statement, however, unlike the if statement, the boolean expression is checked again and again in a loop
    b. This is till the boolean expression is evaluated to ***false***
    c. If the boolean expression is never ***false***, the while loop goes through an **Infinite loop**.

    The syntax is

```
while (true) {
    System.out.println("The number inside a infinite while loop”);
}
```




```java
// Write the program to print the counter from 1 to 5 using while loop
class Counter {
    public static void main(String[] args) {
        // create the counter variable
        int counter = 0;

        // print the counter
        while (counter < 5) {
            counter = counter + 1;
            System.out.println("Counter = " + counter);
        }
    }
}
```

10. **Java for Loop -** The for loop is a finite loop and is made up of 3 expressions, one for initialization of the counter, another for the boolean expression to execute the for block, and third to update the counter.

```
for (initialization; booleanExpresssion; update) {
    // statement(s)
}
```

```java
// Write the program to print the counter from 1 to 5 using while loop
class Counter {
    public static void main(String[] args) {
        // print the counter
        for (int i = 0; i < 5; i++) {
            System.out.println("Counter = " + (i+1));
        }
    }
}
```

11. **Java *break* and *continue* Statement -** The break and continue statements are used inside a loop to alter its flow.
    a. The ***break*** statement terminates the loop immediately when it is encountered.
    b. The ***continue*** statement allows us to skip statements inside the loop.

```java
// Write the program to print the odd counter from 1 to 10 using while loop and
// break if divisible by 7. The output will be 1, 3, and 5
class Counter {
    public static void main(String[] args) {
        // create the counter variable
        int counter = 0;

        // print the counter if odd
        while (counter <= 10) {
            counter = counter + 1;
            if (counter % 2 == 0) continue;
            if (counter % 7 == 0) break;
            System.out.println("Counter = " + counter);
        }
    }
}
```

12. **Java *switch* Statement -** The switch statement starts with the switch keyword followed by a variable or an expression inside the parentheses.
    a. The ***switch*** statement can be replaced by if else statement as well, but the switch statement makes it more readable.
    b. Inside the switch statement, we have multiple ***case*** statements to perform multiple operations.
    c. **Note:** We have used the ***break*** statement after each case to terminate the switch statement once the case is executed.
    d. If the ***break*** statement within the case is not used all the following cases till the break statement is reached will be executed.

Java

```java
// This program takes in a number and prints the day of the week and also
// prints if it is a weekday or weekend
import java.util.Scanner;

class DayOfWeek {
    public static void main(String[] args) {
        // Defining the scanner object to take user input
        Scanner input = new Scanner(System.in);

        // take number input
        System.out.println("Enter a number between 1 to 7: ");
        int weekDayNumber = input.nextInt();

        // Prints the Day of the week corresponding to the case being executed
        // depending on the value of number
        switch (weekDayNumber) {
            case 1:
                System.out.println("Sunday");
                break;
            case 2:
                System.out.println("Monday");
                break;
            case 3:
                System.out.println("Tuesday");
                break;
            case 4:
                System.out.println("Wednesday");
                break;
            case 5:
                System.out.println("Thursday");
                break;
            case 6:
                System.out.println("Friday");
                break;
            case 7:
                System.out.println("Saturday");
                break;
            default:
                System.out.println("Invalid Number");
        }
```

```
        // The following code will check the number is a weekday or the weekend
        switch(weekDayNumber) {
            case 2:
            case 3:
            case 4:
            case 5:
            case 6:
                System.out.println("Weekday");
                break;
            case 7:
            case 1:
                System.out.println("Weekend");
                break;
        }
    }
}
```

Best Programming Practice

1. All values as variables including Fixed, User Inputs, and Results
2. Proper naming conventions for all variables
   ```
   String name = "Eric";
   double height = input.nextDouble();
   double totalDistance = distanceFromToVia + distanceViaToFinalCity;
   ```
3. Proper Program Name and Class Name
4. Follow proper indentation
5. Give comments for every step or logical block like a variable declaration or conditional and loop blocks

1. **Sample Program 1 -** Create a program to check if 3 values are internal angles of a triangle.

   IMP => Follow Good Programming Practice demonstrated below in all Practice Programs

   Hint =>

   a. Get integer input for 3 variables named x, y, and z.
   b. Find the sum of x, y, and z.
   c. If the sum is equal to 180, print `"The given angles are internal angles of a triangle"` else print They are not

Java

```java
// Creating Class with name TriangleChecker indicating the purpose is to
// check if the internal angles add to 180
import java.util.Scanner;

class TriangleChecker {
   public static void main(String[] args) {
      // Create a Scanner Object
      Scanner input = new Scanner(System.in);

      // Get 3 input values for angles
      int x = input.nextInt();
      int y = input.nextInt();
      int z = input.nextInt();

      // Find the sum of all angles
      int sumOfAngles = x + y + z;

      // Check if sum is equal to 180 and print either true or false
      System.out.println("The given angles " +x+ ", " +y+ ", " + z +
                         " add to " + sumOfAngles);

      if (sumOfAngles == 180) {
         System.out.println("The given angles are internal angles of a " +
                            "Triangle");
      } else {
         System.out.println("The given angles are not internal angles " +
                            "of a Triangle");
      }

      // Closing the Scanner Stream
      input.close();
   }
}
```

2. **Sample Program 2 -** Create a program to find the sum of all the digits of a number given by a user.

   Hint =>

   a. Get an integer input for the number variable.
   b. Create an integer variable sum with an initial value of 0.
   c. Create a while loop to access each digit of the number.
   d. Inside the loop, add each digit of the number to the sum.
   e. Finally, print the sum outside the loop

Java

```java
// Create SunOfDigit Class to compute the sum of all digits of a number
import java.util.Scanner;
class SumOfDigits {
   public static void main(String[] args) {
      // Create a Scanner Object
      Scanner input = new Scanner(System.in);

      // Get input value for number
      int origNumber = input.nextInt();

      // Define variable number and sum initialized to zero
      int number = origNumber;
      int sum = 0;

      // Run while loop to access each digit of number
      while (number != 0) {
         // Use number % 10 to find each digit of number from last
         int digit = number % 10;

         // add each digit to sum
         sum += digit;

         // Remove last digit from number essentially get the quotient
         number = number / 10;
      }

      // Print the sum and close the Scanner Stream
      System.out.println("The sum of digit of number:" +origNumber+ " = " +
                         sum);
      input.close();
   }
}
```