Control Structures

Module-1

Dr.M. Sivakumar
Professor & Technical Trainer / CSE
Saveetha School of Engineering, SIMATS, Chennai-602 105
sivakumarm.sse@saveetha.com
+91-9500600868

Outline

- ✓ If statement
- ✓ Two way if statement
- ✓ Nested if statement
- ✓ Switch statement
- ✓ Conditional Expression
- ✓ While loop
- ✓ Do-while loop
- ✓ For loop
- ✓ Break and continue statement

Control Statements

- Control Statements in Java is one of the fundamentals required for Java Programming. It allows the smooth flow of a program.
- Statement can simply be defined as an instruction given to the computer to perform specific operations.
- A control statement in java is a statement that determines whether the other statements will be executed or not.
- Control statements in Java,
  - `If` statement
  - `If-else` statement
  - `If-else ladder` statement
  - `Switch` statement

if statement

- if statement tests the condition. It executes the if block if condition is true.

```
public class IfStatementDemo{
    public static void main(String[] ar)
    {
        int a = 10, b = 20;
        if( a < b ) {
                System.out.println("A is smaller than B");
        }
    }
}
```

if-else statement

- if-else statement also tests the condition. It executes the if block if condition is true otherwise else block is executed.

```
public class IfElseStatementDemo{
    public static void main(String[] ar)
    {
        int a = 10, b = 20;
        if(a<b) {
                System.out.println("A is smaller than B");
        }
        else {
                System.out.println("A is not smaller than B");
        }
    }
}
```

if-else statement

- if-else-if ladder statement executes one condition from multiple statements.

```
int marks = 65;

if (marks < 60) {
        System.out.println("fail");
} else if (marks >= 60 && marks < 80) {
        System.out.println("B grade");
} else if (marks >= 80 && marks < 90) {
        System.out.println("A grade");
} else if (marks >= 90 && marks < 100) {
        System.out.println("A+ grade");
} else {
        System.out.println("Invalid!");
}
```

Nested If statement

- We can also use if/else if statement inside another if/else if statement, this is known as nested if statement.

```
int username = Integer.parseInt(args[0]);
int password = Integer.parseInt(args[1]);
double balance = 123456.25;

if(username==1234){
    if(password==987654){
        System.out.println("Your Balance is ="+balance);
    }
    else{
        System.out.println("Password is invalid");
    }
}
else{
    System.out.println("Username is invalid");
}
```

switch statement

- switch statement executes one statement from multiple conditions. It is like if-else-if ladder statement.

```
public class SwitchExampleDemo {
    public static void main(String[] args)
    {
        int number = 20;
        switch (number) {
            case 10:
                System.out.println("10");
                break;
            case 20:
                System.out.println("20");
                break;
            default:
                System.out.println("Not 10 or 20");
        }
    }
}
```

Programs to perform (Conditional Statements)

- Write a Java program to get a number from the user and print whether it is positive or negative.
- Write a program to find maximum no from given 3 no.
- The marks obtained by a student in 5 different subjects are input through the keyboard.
  - The student gets a division as per the following rules:
    - Percentage above or equals to 60-first division
    - Percentage between 50 to 59-second division
    - Percentage between 40 and 49-Third division
    - Percentage less than 40-fail

  Write a program to calculate the division obtained by the student.
- Write a Java program that takes a number from the user and displays the name of the weekday accordingly (For example if user enter 1 program should return Monday) .

Looping Statement

- Looping in programming languages is a feature which facilitates the execution of a set of instructions/functions repeatedly while some condition evaluates to true.
- Java provides three ways for executing the loops. While all the ways provide similar basic functionality, they differ in their syntax and condition checking time.
  - While loop
  - Do-while loop
  - For
  - Foreach (will cover this after array)

While Loop

- while loop is used to iterate a part of the program several times. while is entry control loop.
- If the number of iteration is not fixed, it is recommended to use while loop.

```
//code will print 1 to 9
public class WhileLoopDemo {
    public static void main(String[] args) {
        int number = 1;
        while(number < 10) {
            System.out.println(number);
            number++;
        }
    }
}
```

Do-while Loop

- do-while loop is executed at least once because condition is checked after loop body.

```
//code will print 1 to 9
public class DoWhileLoopDemo {
    public static void main(String[] args) {
        int number = 1;
        do {
                System.out.println(number);
                number++;
        }while(number < 10) ;
    }
}
```

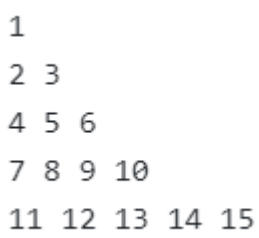
For Loop

- for loop is used to iterate a part of the program several times.
- If the number of iteration is fixed, it is recommended to use for loop.

```
//code will print 1 to 9
public class ForLoopDemo {
    public static void main(String[] args)
    {
        for(int number=1;number<10;number++)
        {
                System.out.println(number);
        }
    }
}
```

Programs to perform (Looping Statements)

- Write a program to print first n odd numbers.
- Write a program to check that the given number is prime or not.
- Write a program to draw given patterns,



Break statement

- When a break statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.

```
//code will print 1 to 4 followed by “After Loop”
public class BreakDemo{
    public static void main(String[] args)
    {
        for(int number=1;number<10;number++)
        {
            if(number==5) {
                break;
            }
            System.out.println(number);
        }
        System.out.println(“After Loop”);
    }
}
```

Continue statement

- The continue statement is used in loop control structure when you need to immediately jump to the next iteration of the loop. It can be used with for loop or while loop.

```
//code will print 1 to 9 but not 5, followed by “After Loop”
public class ContinueDemo {
    public static void main(String[] args)
    {
        for(int number=1;number<10;number++)
        {
            if(number==5) {
                continue;
            }
            System.out.println(number);
        }
        System.out.println(“After Loop”);
    }
}
```