Module 2: Looping Constructs and Arrays

Premanand S

Assistant Professor
School of Electronics Engineering (SENSE)
Vellore Institute of Technology
Chennai Campus

premanand.s@vit.ac.in

January 22, 2025

Topics covered in Module 2,

- Control looping and Constructs
- Arrays
- One-dimensional arrays
- Multi-dimensional arrays
- Enhanced for loop
- Strings
- Wrapper classes

Verdict of Module 2,

- Mastering loops like for, while, and do-while is crucial for automating repetitive tasks.
- Advanced features like break, continue, and else with loops provide fine control over iterations.
- Understanding one-dimensional and multi-dimensional arrays is vital for implementing algorithms and solving mathematical problems efficiently.
- Enhanced loop, Simplifies array and collection traversal, making code cleaner and easier to read.
- Proficiency in string operations is essential for developing user interfaces, parsing data, and handling text in various applications.
- Wrapper classes bridge primitives and object-oriented features, enabling compatibility with advanced frameworks and APIs.

1. Control Looping Constructs

- Loops enable repetitive execution of code blocks.
- They reduce redundancy and enhance code efficiency.
- **Types of Loops in Java:**
  - For Loop
  - While Loop
  - Do-While Loop
- Break, Continue, Pass and else block with loops

For Loop

Syntax:

Example (Understanding)

```
for (initialization; condition; increment/decrement) {
    // Body of the loop
}

Example:
for (int i = 0; i < 5; i++) {
    System.out.println(i);
}
```

While Loop

Syntax:

Example (Understanding)

```
while (condition) {
    // Body of the loop
}

Example:
int i = 0;
while (i < 5) {
    System.out.println(i);
    i++;
}
```

Do While Loop

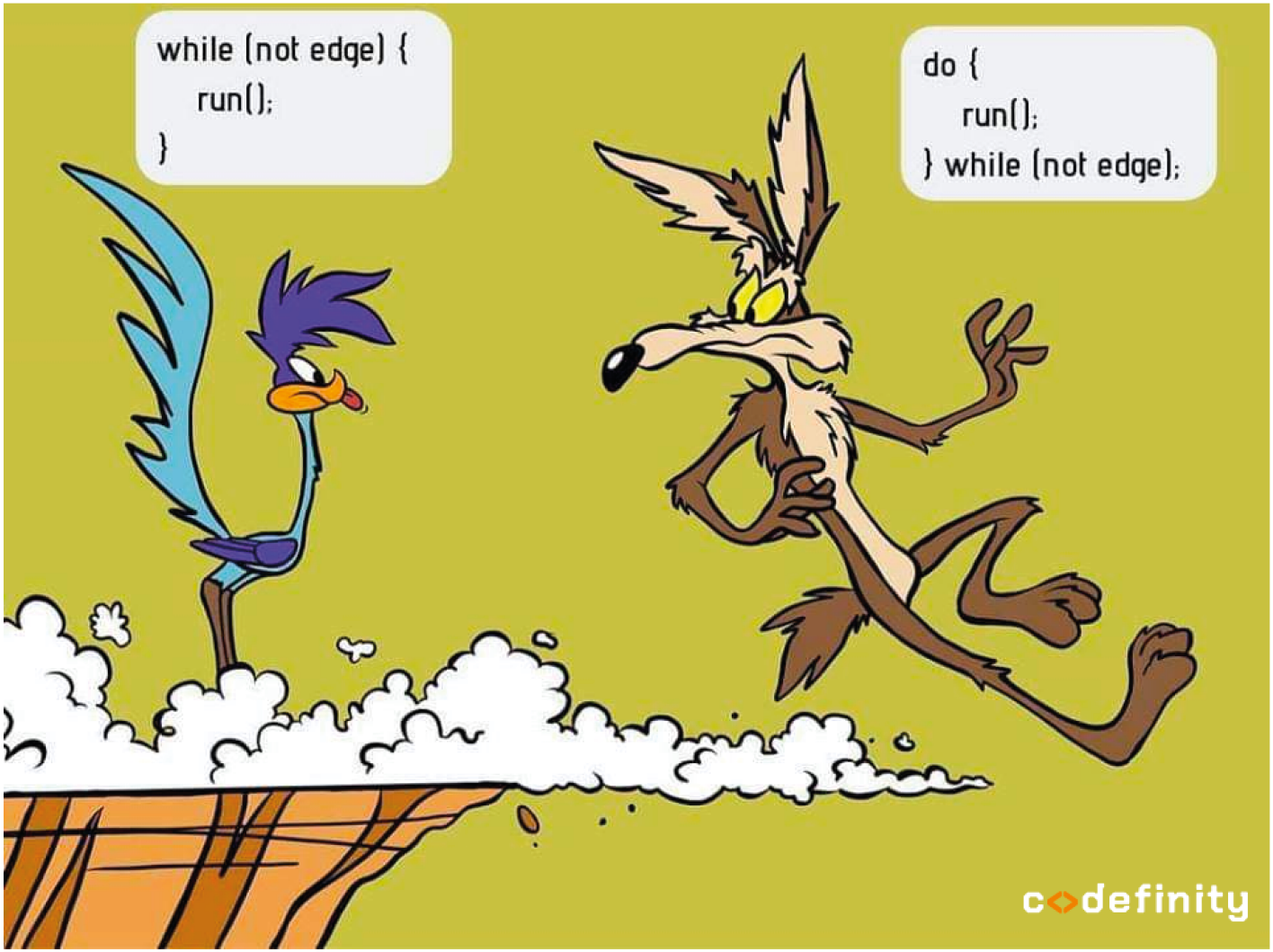
Syntax:

Example (Understanding)

```
do {
    // Body of the loop
} while (condition);

Example:
int i = 0;
do {
    System.out.println(i);
    i++;
} while (i < 5);
```

While Vs Do-While loops



Key Elements of Loops

- **Initialization:** Sets the starting value of the loop variable.
- **Condition:** Logical test to determine if the loop continues or stops.
- **Increment/Decrement:** Updates the loop variable to progress iteration.

Break Statement

Usage: Exits the loop immediately. **Example:**

Example (Understanding)

```
for (int i = 0; i < 10; i++) {
    if (i == 5) {
        break;
    }
    System.out.println(i);
}
```

Continue Statement

Usage: Skips the current iteration. **Example:**

Example (Understanding)

```
for (int i = 0; i < 10; i++) {
    if (i % 2 == 0) {
        continue;
    }
    System.out.println(i);
}
```

Pass Equivalent

Usage: Java uses an empty statement (‘;‘) as a placeholder. **Example:**

Example (Understanding)

```
for (int i = 0; i < 5; i++) {
    if (i == 3) {
        ; // Empty statement
    }
    System.out.println(i);
}
```

Question: Predict the Output

What will be the output of the following code?

- Modify the condition to `i <5` and observe the changes.

Code:

Example (Understanding)

```
public class Main
{
 public static void main(String[] args) {
  for (int i = 1; i <= 5; i++) {
    System.out.print(i + " ");
}
 }
}
```

Answer: Predict the Output

Code:

Example (Understanding)

```
Output:
1 2 3 4 5
```

Fill in the Blanks: Print the First 10 Even Numbers

Complete the loop to print the first 10 even numbers:
Code:

Example (Understanding)

```
for (int i = ___; i <= ___; i += ___) {
    System.out.println(i);
}
```

Fill in the Blanks: Print the First 10 Even Numbers

Code:

Example (Understanding)

```
public class Main
{
    public static void main(String[] args) {
        for (int i = 2; i <= 20; i += 2) {
            System.out.println(i);
        }
    }
}
Output:
2 4 6 8 10 12 14 16 18 20
```

Question: Counting Digits

Write a program to count the number of digits in a given integer using a while loop.

Answer: Counting Digits

Example (Understanding)

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        int number = scanner.nextInt();
        int count = 0; // Initialize digit count to 0
        int temp = Math.abs(number); // Handle negative no

        while (temp > 0) {
            temp /= 10; // Remove the last digit
            count++;    // Increment the digit count
        }
```

Answer: Counting Digits (Contd...)

Example (Understanding)

```
        // Handle the case when the number is 0
        if (number == 0) {
            count = 1;
        }

        System.out.println("The number of digits in the
        given integer is: " + count);
    }
}
```

Question: Reverse a String

Use a for loop to reverse the string "Hello".

Answer: Reverse a String

Example (Understanding)

```
public class Main {
    public static void main(String[] args) {
        String original = "Hello";
        String reversed = "";

        // Loop through the string in reverse order
        for (int i = original.length() - 1; i >= 0; i--) {
        // Append each character to the reversed string
            reversed += original.charAt(i);
        }

        System.out.println("Original string: " + original);
        System.out.println("Reversed string: " + reversed);
    }}
```

Question: Pattern Printing

Print the following pattern using nested for loops

Example (Understanding)

```
*
**
***
****
*****
```

Answer: Pattern Printing

Example (Understanding)

```
public class Main {
    public static void main(String[] args) {
        int rows = 5; // Number of rows in the pattern
        // Outer loop for rows
        for (int i = 1; i <= rows; i++) {
        // Inner loop for columns
            for (int j = 1; j <= i; j++) {
                System.out.print("*");
            }
            // Move to the next line after each row
            System.out.println();
        }    } }
```

Question: Pattern Printing

Print the following pattern using nested for loops

Example (Understanding)

```
1
12
123
1234
12345
```

Answer: Pattern Printing

Example (Understanding)

```
public class Main {
    public static void main(String[] args) {
        int rows = 5;
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(j);
            }
            System.out.println();
        }
    }
}
```

Question: Pattern Printing

Print the following pattern using nested for loops

Example (Understanding)

```
A
AB
ABC
ABCD
ABCDE
```

Answer: Pattern Printing

Example (Understanding)

```
public class Main {
    public static void main(String[] args) {
        int rows = 5;
        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print((char)('A' + j - 1));
            }
            System.out.println();
        }
    }
}
```

Question: Pattern Printing

Print the following pattern using nested for loops

Example (Understanding)

*****

****

***

**

*

Answer: Pattern Printing

Example (Understanding)

```
public class Main {
    public static void main(String[] args) {
        int rows = 5;
        for (int i = rows; i >= 1; i--) {
            for (int j = 1; j <= i; j++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

Question: Pattern Printing

Print the following pattern using nested for loops

Example (Understanding)

```
    *
   ***
  *****
 *******
*********
```

Answer: Pattern Printing

Example (Understanding)

```
public class Main {
    public static void main(String[] args) {
        int rows = 5;
        for (int i = 1; i <= rows; i++) {
            for (int j = i; j < rows; j++) {
                System.out.print(" ");
            }
            for (int k = 1; k <= (2 * i - 1); k++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

Question: Pattern Printing

Print the following pattern using nested for loops

Example (Understanding)

```
    *
   ***
  *****
 *******
*********
 *******
  *****
   ***
    *
```

Answer: Pattern Printing

Example (Understanding)

```
public class Main {
    public static void main(String[] args) {
        int rows = 5;
        // Upper half
        for (int i = 1; i <= rows; i++) {
            for (int j = i; j < rows; j++) {
                System.out.print(" ");
            }
            for (int k = 1; k <= (2 * i - 1); k++) {
                System.out.print("*");
            }
            System.out.println();
        }
```

Answer: Pattern Printing (Contd...)

Example (Understanding)

```
        // Lower half
        for (int i = rows - 1; i >= 1; i--) {
            for (int j = rows; j > i; j--) {
                System.out.print(" ");
            }
            for (int k = 1; k <= (2 * i - 1); k++) {
                System.out.print("*");
            }
            System.out.println();
        }
    }
}
```

Question: Pattern Printing

Print the following pattern using nested for loops

Example (Understanding)

```
*****
*   *
*   *
*   *
*****
```

Answer: Pattern Printing

Example (Understanding)

```
public class Main {
    public static void main(String[] args) {
        int size = 5;
        for (int i = 1; i <= size; i++) {
            for (int j = 1; j <= size; j++) {
                if (i == 1 || i == size ||
                j == 1 || j == size) {
                    System.out.print("*");
                } else {
                    System.out.print(" ");
                }               }
            System.out.println();
        }    } }
```

Question: Pattern Printing (Pascal's Triangle)

Print the following pattern using nested for loops

Example (Understanding)

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

Answer: Pattern Printing

Example (Understanding)

```
public class Main {
    public static void main(String[] args) {
        int rows = 5;
        for (int i = 0; i < rows; i++) {
            for (int j = 1; j <= rows - i; j++) {
                System.out.print(" ");
            }
            int num = 1;
            for (int j = 0; j <= i; j++) {
                System.out.print(num + " ");
                num = num * (i - j) / (j + 1);
            }
            System.out.println();
        }
    }
}
```

Question: Prime Number Checker

Write a program to check if a given number is prime using a for loop.

Code: Prime Number Checker

Syntax:

Example (Understanding)

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number to check if
        it's prime: ");
        int number = scanner.nextInt();
        if (isPrime(number)) {
            System.out.println(number +
            " is a prime number.");
        } else {
            System.out.println(number +
            " is not a prime number.");
        }    }
```

Code: Prime Number Checker (Contd...)

Example (Understanding)

```
public static boolean isPrime(int num) {
    if (num <= 1) {
    // Numbers less than or equal to 1 are not prime.
        return false;
    }

    for (int i = 2; i <= Math.sqrt(num); i++) {
    // Check divisors up to the square root of num.
        if (num % i == 0) {
        // If divisible by any number, it's not prime.
            return false;
        }         }
    // If no divisors are found, it's prime.
    return true;
} }
```

Question: Login Attempts

Create a login system where the user gets three attempts to enter the correct password.

Code: Login Attempts

Example (Understanding)

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String correctPassword = "secure123";
        int attempts = 3;

        System.out.println("Welcome to the Login System");
        while (attempts > 0) {
            System.out.print("Enter your password: ");
            String inputPassword = scanner.nextLine();
```

Code: Login Attempts (Contd...)

Example (Understanding)

```
            if (inputPassword.equals(correctPassword)) {
                System.out.println("Login successful!");
                return;
            } else {
                attempts--;
                if (attempts > 0) {
                    System.out.println("Incorrect password.
                    You have " + attempts + " attempts left.");
                } else {
                    System.out.println("Incorrect password.
                    You have been locked out.");
                }               }       }
        scanner.close();
    }}
```