Flow control statements in Java

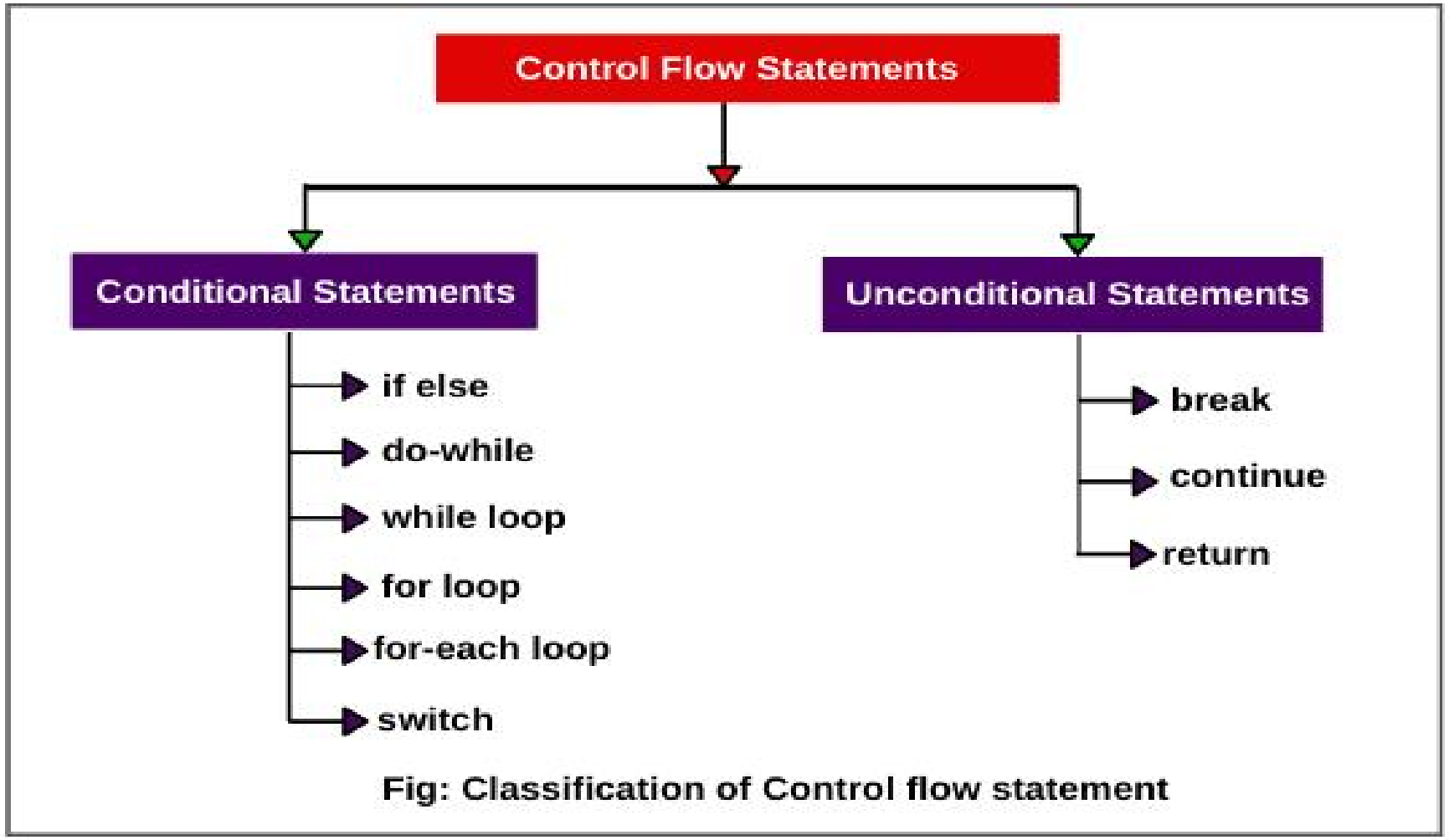


Fig: Classification of Control flow statement

Decision-Making Statements or Conditional Statements

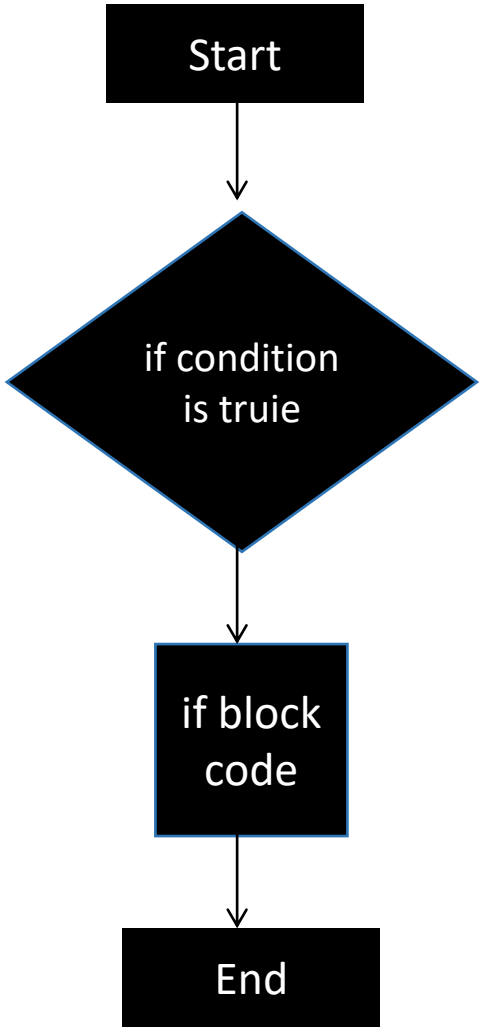
- if Statement
- if-else statement
- if-else-if statement
- switch statement

if Statement

Syntax:

```
if (condition) {
    // code to be executed if condition is true
}
```

Flow Diagram for if



Real Time Example for if:

Condition

Action

```
1  if (it is raining) {
2      use an umbrella
3  }
```

if Statement Example

```
public class Main {
    public static void main(String[] args) {
        int number = 10;

        if (number > 0) {
            System.out.println("The number is positive.");
        }
    }
}
```

output:

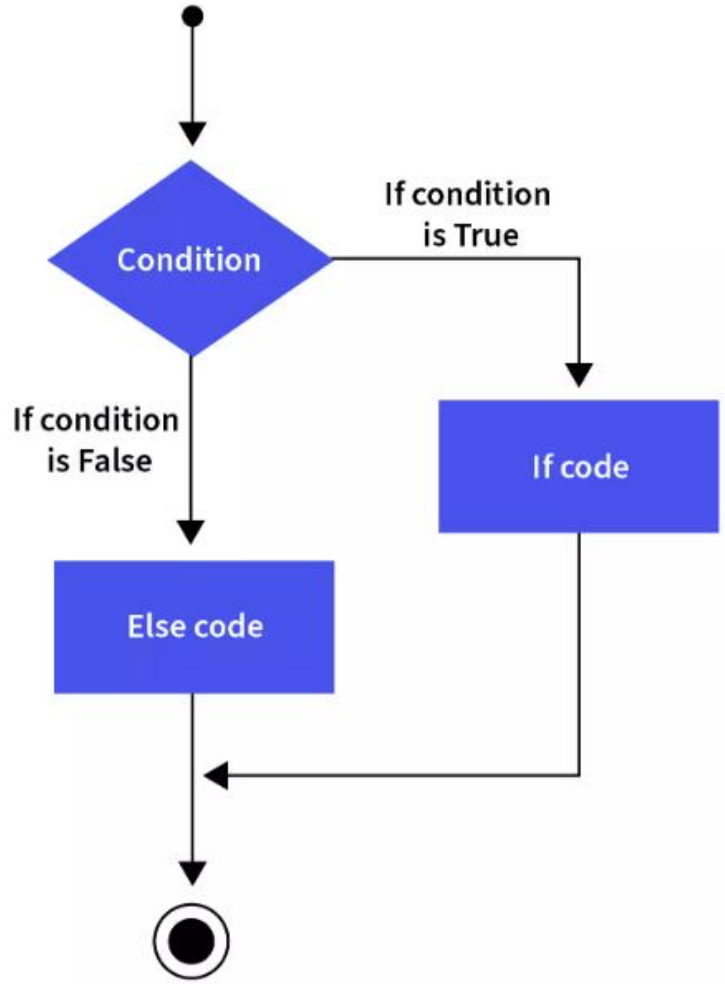
The number is positive.

if-else statement

Syntax:

```
if (condition) {
    // code to be executed if condition is true
} else {
    // code to be executed if condition is false
}
```

Flow Diagram for if else



if else Statement Example

```
public class Main {
    public static void main(String[] args) {
        int number = -5;

        if (number > 0) {
            System.out.println("The number is positive.");
        } else {
            System.out.println("The number is not positive.");
        }
    }
}
```

output:

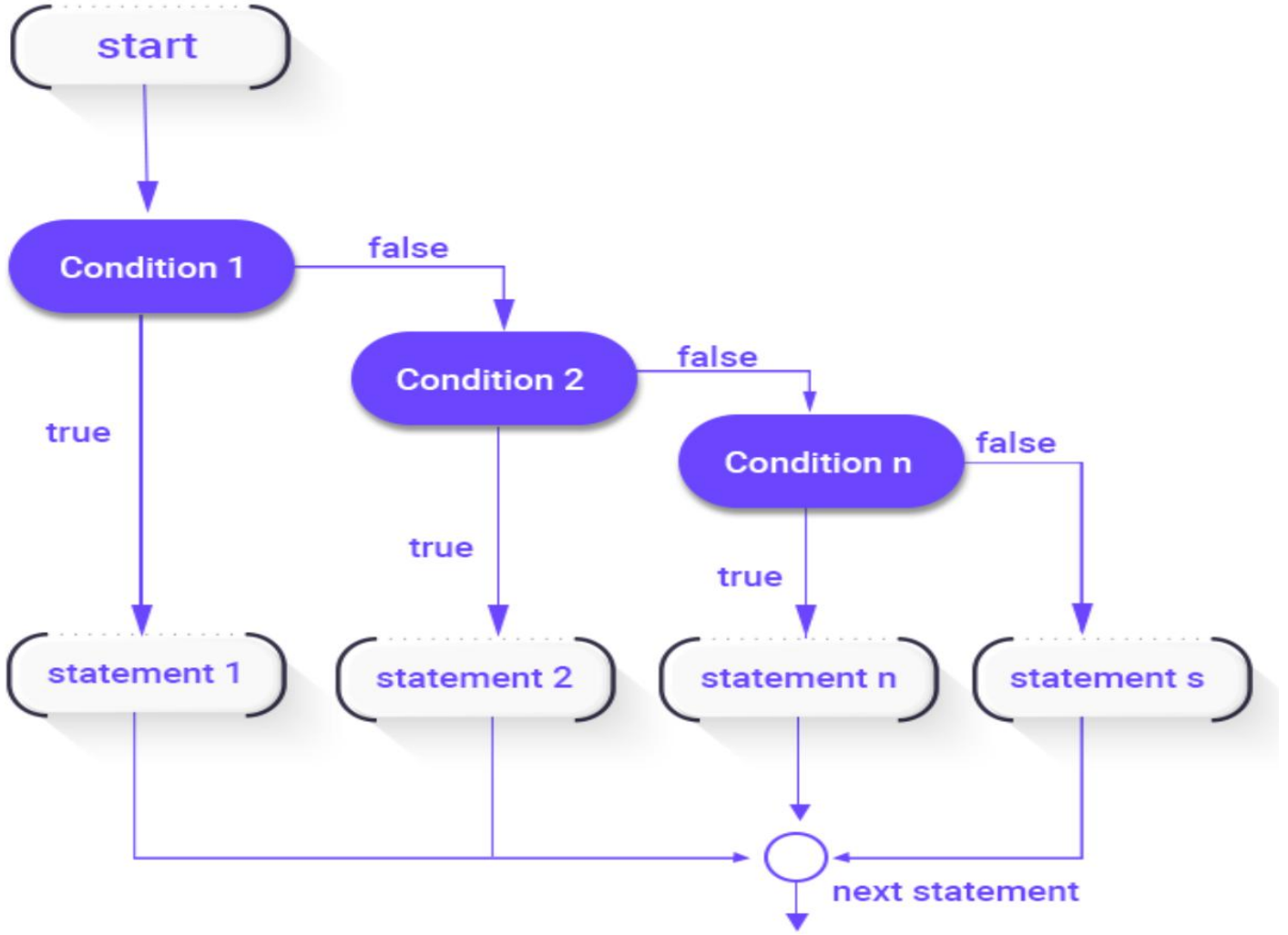
The number is not positive.

if-else-if ladder

Syntax:

```
if (condition1) {
    // code to be executed if condition1 is true
} else if (condition2) {
    // code to be executed if condition2 is true
} else {
    // code to be executed both conditions are false
}
```

Flow Diagram for if-else-if ladder



if-else-if ladder Example

```
public class Main {
    public static void main(String[] args) {
        int hour = 14; // Let's assume 24-hour format (2 PM)

        if (hour < 12) {
            System.out.println("Good morning!");
        } else if (hour < 17) {
            System.out.println("Good afternoon!");
        } else if (hour < 21) {
            System.out.println("Good evening!");
        } else {
            System.out.println("Good night!");
        }

        System.out.println("Have a nice day!");
    }
}
```

Output:

Good afternoon!

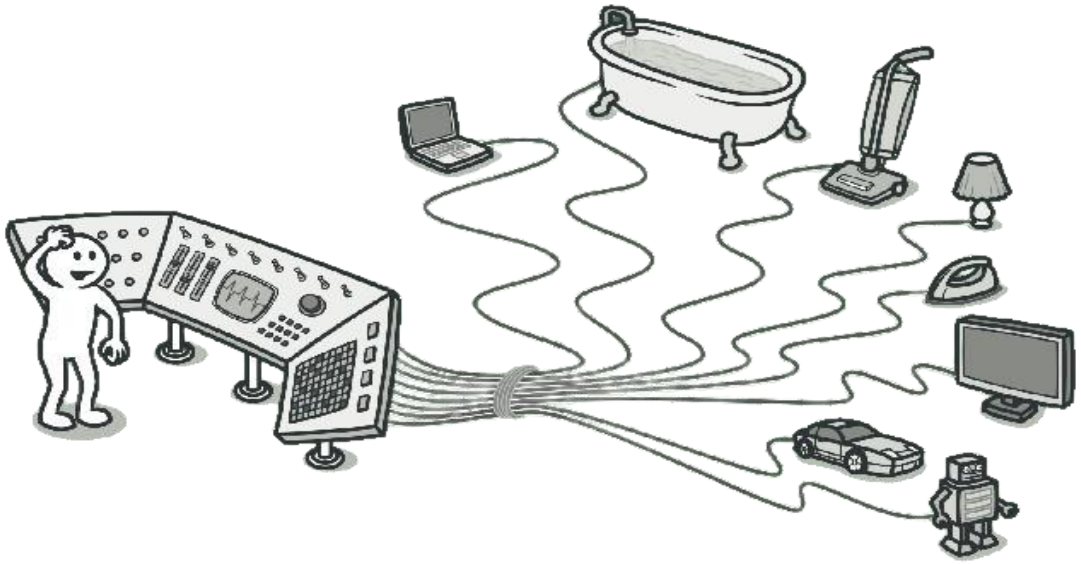
Have a nice day!

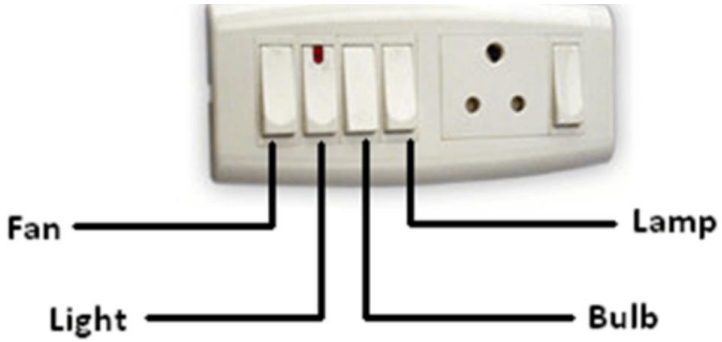
switch statement

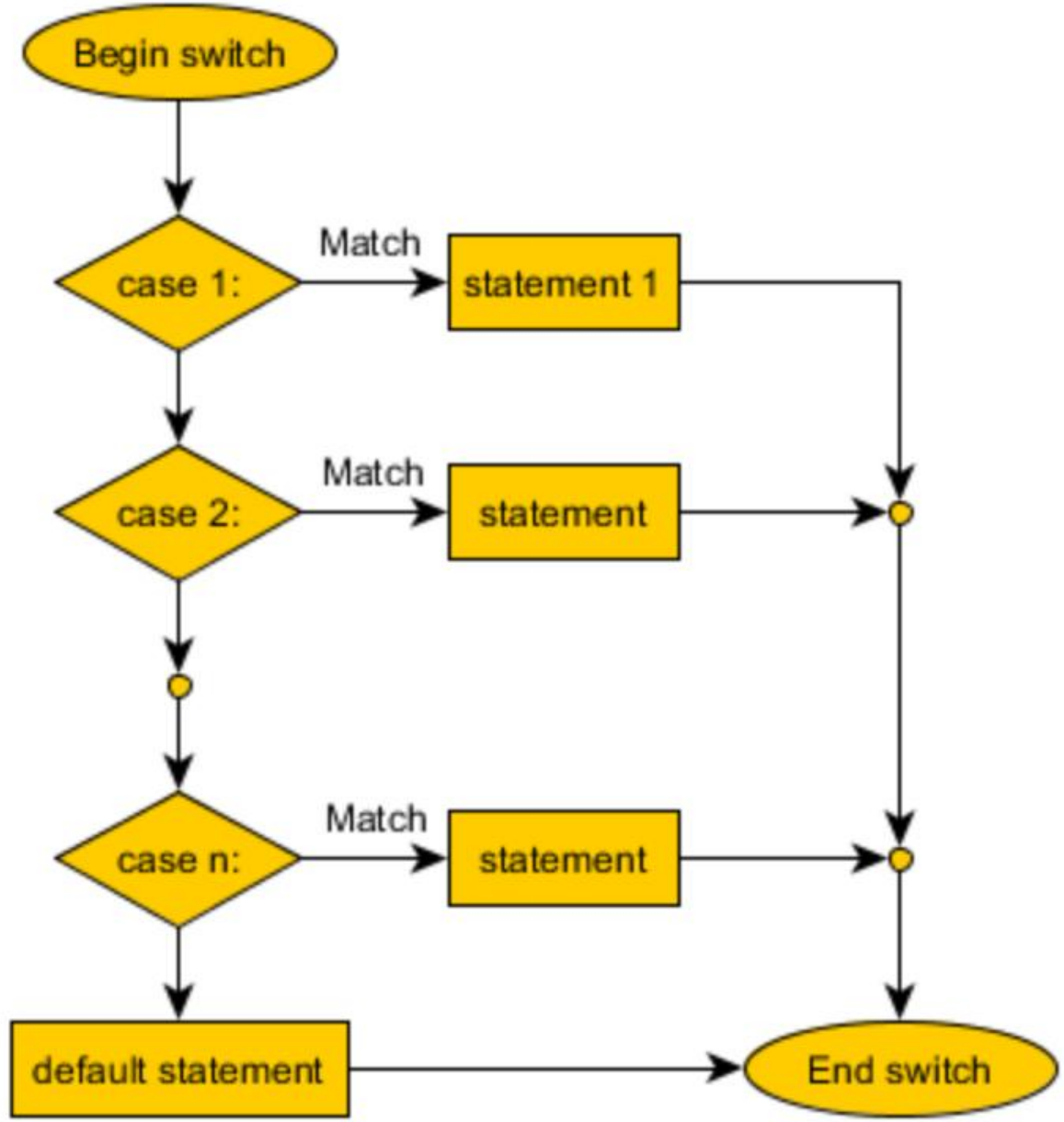
Syntax:

```
switch (expression) {
  case value1:
    // code to be executed if expression equals value1
    break;
  case value2:
    // code to be executed if expression equals value2
    break;
  // you can have any number of case statements
  default:
    // code to be executed if expression doesn't match any case
}
```

switch statement







switch Statement Example

```
public class SwitchVowelExample {
public static void main(String[] args) {
   char ch='O';
   switch(ch)
   {
      case 'a':
         System.out.println("Vowel");
         break;
      case 'e':
         System.out.println("Vowel");
         break;
      case 'i':
         System.out.println("Vowel");
         break;
      case 'o':
         System.out.println("Vowel");
         break;
      case 'u':
         System.out.println("Vowel");
         break;
      case 'A':
         System.out.println("Vowel");
         break;
      case 'E':
         System.out.println("Vowel");
         break;
      case 'I':
         System.out.println("Vowel");
         break;
      case 'O':
         System.out.println("Vowel");
         break;
      case 'U':
         System.out.println("Vowel");
         break;
      default:
         System.out.println("Consonant");
   }
}
}
```

Output:
Vowel

Loop or Iterative Statements

- for Loop
- while Loop
- do-while Loop
- for-each Loop

for loop

Syntax:

```
for (initialization; condition; update) {
    // code to be executed
}
```

for loop Example - 1

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

Output

1
2
3
4
5

for loop Example - 2

```
public class Main {
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i += 2) {
            System.out.println(i);
        }
    }
}
```

Output

```
1
3
5
7
9
```

Pattern 1: Square Pattern

```
public class Main {
    public static void main(String[] args) {
        int rows = 5;

        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= rows; j++) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

Output

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

Pattern 2: Right Triangle Pattern

```
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("* ");
            }
            System.out.println();
        }
    }
}
```

Output

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

Pattern 3: Pyramid Pattern

```
public class Main {
    public static void main(String[] args) {
        int rows = 5;

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

Output

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

while loop

Syntax:

```
while (condition) {
    // code to be executed
}
```

while loop Example - 1

```
public class Main {
    public static void main(String[] args) {
        int i = 1;

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

Output

1
2
3
4
5

while loop Example - 2

```
public class Main {
    public static void main(String[] args) {
        int i = 1;

        while (i <= 5) {
            System.out.println(i);
            i+=2;
        }
    }
}
```

Output

1

3

5

do-while loop

Syntax:

```
do {
    // code to be executed
} while (condition);
```

The do-while loop is similar to the while loop, but with one key difference: the do-while loop executes its body **at least once,** even if the condition is initially false.

do while loop Example

```
public class Main {
    public static void main(String[] args) {
        int i = 1;

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

Output

1
2
3
4
5

Enhanced for loop (for-each loop)

Syntax:

```
for (type element : array) {
    // code to be executed
}
```

for-each loop Example

```
public class Main {
    public static void main(String[] args) {
        int[] numbers = {1, 2, 3, 4, 5};

        for (int number : numbers  {
            System.out.println(number);
        }
    }
}
```

Output

1
2
3
4
5

Flow Control or Jump Statements

- continue
- break

break statement

Syntax:

```
for (int i = 0; i < 10; i++) {
    if (i == 5) {
        break; // exit the loop
    }
    // code to be executed
}
```

break statement Example:

```
public class Main {
    public static void main(String[] args) {
        int target = 5;

        for (int i = 1; i <= 10; i++) {
            if (i == target) {
                System.out.println("Target number found: " + i);
                break;
            }
        }
    }
}
```

Output

Target number found: 5

continue statement

Syntax:

```
for (int i = 0; i < 10; i++) {
    if (i == 5) {
        continue; // skip the rest of the loop iteration
    }
    // code to be executed
}
```

continue statement Example - 1

```
public class Main {
    public static void main(String[] args) {
        for (int i = 1; i <= 5; i++) {
            if (i % 2 == 0) {
                continue;
            }
            System.out.println(i);
        }
    }
}
```

Output

1
3
5

continue statement Example - 2

```
public class Main {
    public static void main(String[] args) {
        for (int i = 1; i <= 5; i++) {
            if (i % 2 == 1) {
                continue;
            }
            System.out.println(i);
        }
    }
}
```

Output

2

4