OBJECT ORIENTED PROGRAMMING USING JAVA

OUTLINE

- Conditional Statements
- IF statement
- Switch

CONDITIONAL STATEMENTS

- A conditional statement allows us to choose which statement will be executed. So, it is sometimes called as selection statement.
- Conditional statement gives us the power to make decision.
- Java has the following conditional statements:
  - ❖If
  - ❖if-else
  - ❖nested-if
  - ❖if-else-if
  - ❖switch-case

IF STATEMENT

- if statement is the most simple decision making statement.
- It is used to decide whether a certain statement or block of statements will be executed.
- Syntax:

```
if(condition)
{
    Statement 1;
    Statement 2;
}
```

IF STATEMENT (CONT.…)

- if is a Java reserved word.
- The condition must be a Boolean expression. It must evaluate to either true or false.
- If we do not give the curly braces '{' and '}' after if(condition), then, by default if statement considers the immediate statement inside its block.
- Example:

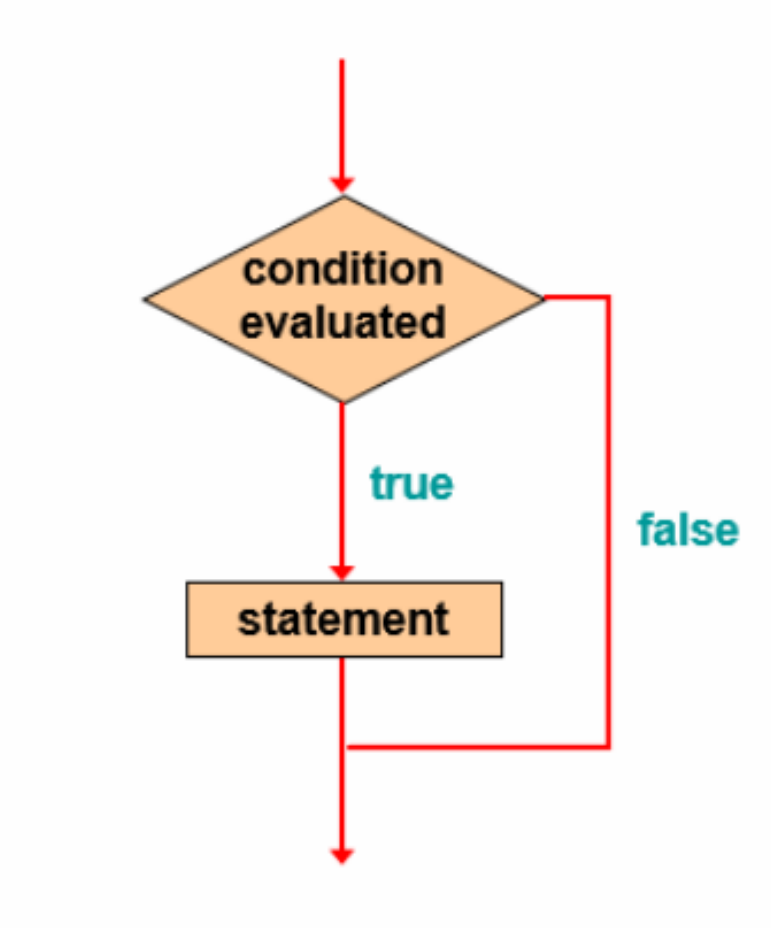
```
if(condition)
statement 1; //Statement 1 will be executed
statement 2;
```

IF STATEMENT (CONT.…)

- if is a Java reserved word.
- The condition must be a Boolean expression. It must evaluate to either true or false.
- If we do not give the curly braces ‘{’ and ‘}’ after if(condition), then, by default if statement considers the immediate statement inside its block.
- Example:

```
if(condition)
statement 1; //Statement 1 will be executed
statement 2;
```

IF STATEMENT (CONT….)



IF STATEMENT (CONT….)

```
public class MyClass {
  public static void main(String[] args) {
    if (20 > 18) {
      System.out.println("20 is greater than 18"); // obviously
    }
  }
}
```

ELSE STATEMENT

- Use the else statement to specify a block of code to be executed, if the condition is false.
- **Syntax:**

```
if(condition)
{
    Statement 1;     // block of code to be executed, if the condition is true
}
else
{
    Statement 2;     // block of code to be executed, if the condition is false
}
```

ELSE STATEMENT

```
public class MyClass {
  public static void main(String[] args) {
    int time = 20;
    if (time < 18) {
      System.out.println("Good day.");
    } else {
      System.out.println("Good evening.");
    }
  }
}
```

ELSE STATEMENT

```
public class MyClass {
  public static void main(String[] args) {
    int time = 20;
    if (time < 18) {
      System.out.println("Good day.");
    } else {
      System.out.println("Good evening.");
    }
  }
}
```

Output
Good evening.

ELSE-IF STATEMENT

- Use the else if statement to specify a new condition if the first condition is false.
- Syntax:

```
if(condition1)
{
        Statement 1; // block of code to be executed, if the condition1 is true
}
else if (condition 2)
{
        Statement 2; // block of code to be executed, if the condition1 is false and condition2 is true
}
else
{
        Statement 3; // block of code to be executed, if the condition1 is false and condition2 is false
}
```

ELSE-IF STATEMENT (CONT.…)

```
public class MyClass {
  public static void main(String[] args) {
    int time = 22;
    if (time < 10) {
      System.out.println("Good morning.");
    } else if (time < 20) {
      System.out.println("Good day.");
    }  else {
      System.out.println("Good evening.");
    }
  }
}
```

ELSE-IF STATEMENT (CONT.…)

```
public class MyClass {
  public static void main(String[] args) {
    int time = 22;
    if (time < 10) {
      System.out.println("Good morning.");
    } else if (time < 20) {
      System.out.println("Good day.");
    }  else {
      System.out.println("Good evening.");
    }
  }
}
```

Output
Good evening.

SWITCH STATEMENT

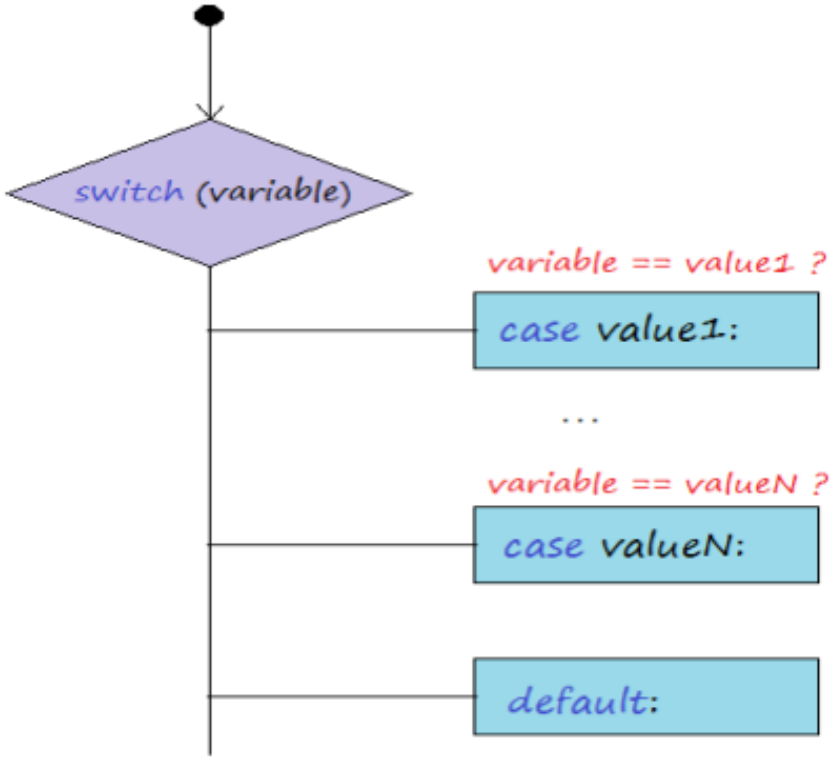
- The switch statement is a multi-way branch statement.
- It provides an easy way to dispatch execution to different parts of code based on the value of the expression.
- The switch expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.
- The break and default keywords are optional.

SWITCH STATEMENT (CONT…)

- Syntax:

```
switch(expression)
{
    case x:
        Statement 1; // code block
        break;
    case y:
        Statement 2; // code block
        break;
    default:
        Statement 3;  // code block
}
```

SWITCH STATEMENT (CONT…)



SWITCH STATEMENT (CONT…)

- Break Keyword
  - ❖ When Java reaches a break keyword, it breaks out of the switch block.
  - ❖ This will stop the execution of more code and case testing inside the block.
  - ❖ When a match is found, and the job is done, it's time for a break.
  - ❖ There is no need for more testing.

SWITCH STATEMENT (CONT.…)

- Default Keyword
  - ❖ The default keyword specifies some code to run, if there is no case match.

SWITCH STATEMENT (CONT.…)

- Some important rule:
  - ❖ The default keyword specifies some code to run, if there is no case match.
  - ❖ The value for a case must be of the same data type as the variable in the switch.
  - ❖ The value for a case must be a constant or a literal. Variables are not allowed.
  - ❖ The break statement is used inside the switch to terminate a statement sequence.
  - ❖ The break statement is optional. If omitted, execution will continue on into the next case.
  - ❖ The default statement is optional and can appear anywhere inside the switch block. In case, if it is not at the end, then, a break statement must be kept after the default statement to omit the execution of the next case statement.

SWITCH STATEMENT (CONT.…)

```
public class MyClass {
 public static void main(String[] args)
{
    int day = 4;
    switch (day)
    {
        case 1:
            System.out.println("Monday");
            break;
       case 2:
             System.out.println("Wednesday");
             break;
        case 3:
             System.out.println("Thursday");
             break;
        case 4:
             System.out.println("Friday");
             break;
         case 5:
             System.out.println("Saturday");
             break;
         case 6:
             System.out.println("Sunday");
            break;
  }
} }
```

SWITCH STATEMENT (CONT.…)

```
public class MyClass {
 public static void main(String[] args)
{
    int day = 4;
    switch (day)
    {
        case 1:
            System.out.println("Monday");
            break;
       case 2:
             System.out.println("Wednesday");
             break;
        case 3:
             System.out.println("Thursday");
             break;
        case 4:
             System.out.println("Friday");
             break;
         case 5:
             System.out.println("Saturday");
             break;
         case 6:
             System.out.println("Sunday");
            break;
  }
} }
```

Output:
Friday

SWITCH STATEMENT (CONT.…)

```
public class MyClass {
  public static void main(String[] args) {
    int day = 4;
    switch (day) {
      case 6:
        System.out.println("Today is Saturday");
        break;
      case 7:
        System.out.println("Today is Sunday");
        break;
      default:
        System.out.println("Looking forward to the Weekend");
    }
  }
}
```

SWITCH STATEMENT (CONT.…)

```
public class MyClass {
  public static void main(String[] args) {
    int day = 4;
    switch (day) {
      case 6:
        System.out.println("Today is Saturday");
        break;
      case 7:
        System.out.println("Today is Sunday");
        break;
      default:
        System.out.println("Looking forward to the Weekend");
    }
  }
}
```

Output
Looking forward to the Weekend

THANK YOU
?