

Lecture – 4 (Control Statement Part - 1)

Java IF Statement

The Java if statement tests the condition. It executes the if block if condition is true.

Syntax:

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

Example:

```
public class IfExample {  
    public static void main(String[] args) {  
        int age=20;  
        if(age>18){  
            System.out.print("Age is greater than 18");  
        }  
    }  
}
```

Output:

Age is greater than 18

Java IF-else Statement

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

Syntax:

```
if(condition){  
    //code if condition is true  
}else{
```

```
//code if condition is false  
}
```

Example:

```
public class IfElseExample {  
    public static void main(String[] args) {  
        int number=13;  
        if(number%2==0){  
            System.out.println("even number");  
        }else{  
            System.out.println("odd number");  
        }  
    }  
}
```

Output:

odd number

Java IF-else-if ladder Statement

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

Syntax:

```
if(condition1){  
    //code to be executed if condition1 is true  
}else if(condition2){  
    //code to be executed if condition2 is true  
}  
else if(condition3){  
    //code to be executed if condition3 is true  
}  
...  
else{  
    //code to be executed if all the conditions are false  
}
```

}

Example:

```
public class IfElseExample {  
    public static void main(String[] args) {  
        int marks=65;  
        if(marks<50){  
            System.out.println("fail");  
        }  
        else if(marks>=50 && marks<60){  
            System.out.println("D grade");  
        }  
        else if(marks>=60 && marks<70){  
            System.out.println("C grade");  
        }  
        else if(marks>=70 && 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!");  
        }  
    }  
}
```

Output:

C grade

Java Switch Statement

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

Syntax:

```
switch(expression){  
  case value1:  
    //code to be executed;  
    break; //optional  
  case value2:  
    //code to be executed;  
    break; //optional  
  .....  
  
  default:  
    code to be executed if all cases are not matched;  
}
```

Example:

```
public class SwitchExample {  
  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;  
      case 30:  
        System.out.println("30");  
        break;  
      default:  
        System.out.println("Not in 10, 20 or 30");  
    }  
  }  
}
```

Output:

20

Java Switch Statement is fall-through

The java switch statement is fall-through. It means it executes all statement after first match if break statement is not used with switch cases.

Example:

```
public class SwitchExample2 {  
    public static void main(String[] args) {  
        int number=20;  
        switch(number){  
            case 10: System.out.println("10");  
            case 20: System.out.println("20");  
            case 30: System.out.println("30");  
            default: System.out.println("Not in 10, 20 or 30");  
        }  
    }  
}
```

Output:

20

30

Not in 10, 20 or 30