Conditional and Jump Statements

January 2, 2025

[1]:
```
// If Statement
val temperature = 15

if (temperature >= 20) {
  println("It's a warm day.")
} else {
  println("It's a cold day.")
}
```

```
It's a cold day.
```

```
temperature = 15
```

[1]: 15

[2]:
```
// If-Else Chained
val grade = 75

if (grade >= 90) {
  println("Excellent!")
} else if (grade >= 75) {
  println("Good Job!")
} else if (grade >= 50) {
  println("Needs Improvement.")
} else {
  println("Fail.")
}
```

```
Good Job!
```

```
grade = 75
```

[2]: 75

[3]:
```
val language = "Scala"

language match {
  case "Java" => println("The language is Java.")
```

```
  case "Scala" => println("The language is Scala.")
  case "Python" => println("The language is Python.")
  case _ => println("Unknown language.")
}
```

```
The language is Scala.
```

```
language = Scala
```

[3]:
```
Scala
```

////////////////////////////// HERE ARE THE JUMP OPERATIONS: //////////////////////////////

[4]:
```
// Return Statement
def multiply(a: Int, b: Int): Int = {
  return a * b // Returning the product
}

val product = multiply(6, 7)
println(s"Product: $product")
```

```
Product: 42
```

```
product = 42


multiply: (a: Int, b: Int)Int
```

[4]:
```
42
```

[5]:
```
import scala.util.control.Breaks._

breakable {
  for (i <- 1 to 10) {
    if (i == 7) break() // Exit the loop when i is 7
    println(s"Current number: $i")
  }
}
```

```
Current number: 1
Current number: 2
Current number: 3
Current number: 4
Current number: 5
Current number: 6
```

[6]:
```
// Do-While Loop
var counter = 3
```

```
do {
  println(s"Counter value: $counter")
  counter -= 1
} while (counter > 0)
```

```
Counter value: 3
Counter value: 2
Counter value: 1
```

```
counter = 0
```

[6]: 0

[ ]: