**Control Flow Statements**

The statements inside your source files are generally executed from top to bottom, in the order that they appear. Control flow statements, however, break up the flow of execution by employing decision making, looping, and branching, enabling your program to conditionally execute particular blocks of code. This section describes the decision-making statements (if-then, if-then-else, switch), the looping statements (for, while, do-while), and the branching statements (break, continue, return) supported by the Java programming language.

**If .. Statement**

public static void main(String[] args) {

 int number = 10;

 if (number > 0) {
 System.out.println("Number is positive.");
 }
 System.out.println("This statement is always executed.");
 }

**If .. else Statement**

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

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

 System.out.println("This statement is always executed.");
 }

**If .. else .. if Statement**

 public static void main(String[] args) {

 int number = 0;

 if (number > 0) {
 System.out.println("Number is positive.");
 }
 else if (number < 0) {
 System.out.println("Number is negative.");
 }
 else {
 System.out.println("Number is 0.");
 }
 }

**Example**

public static void main(String[] args) {

 int TestResult = 76;
 char grade;

 if (TestResult >= 90) {
 grade = 'A';
 } else if (TestResult >= 80) {
 grade = 'B';
 } else if (TestResult >= 70) {
 grade = 'C';
 } else if (TestResult >= 60) {
 grade = 'D';
 } else {
 grade = 'F';
 }
 System.out.println("Grade = " + grade);
 }