Control statements-

This is the most fundamental concepts required for java programmer. It allows smooth flow of execution of program. It controls the flow of program.

There are four types of control statements in java

1. If statements
2. If else statements
3. if-else-if ladder statement
4. Nested if statements
5. Switch statements

1. If statements-

If statement is true then if block is executed.

Syntax-

```
if (Condition) {

Statement 1;

}
Statement 2;
```

Example-

```
public class Arithmatic {

	public static void main(String[] args) {

		int a = 10;
		if (a > 50) {
			System.out.println("a is greater.");
		}
	}
}
```

2. If else statements

   If statement is true then if block is executed, if statement is false then else block is executed.

   Syntax

   if (condition) {

   Statement 1;

   }

   else {

   Statement 2;

   }

```
public class Arithmatic {

	public static void main(String[] args) {

		int a = 10;
		if (a > 50) {
			System.out.println("a is greater.");
		}
		else {
			System.out.println("a is smaller.");
		}
	}
}
```

3. if-else-if ladder statement

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

   Syntax-

```
        if (Condition 1) {

         //executed if condition 1 is true
        }

        else if (Condition 2) {

        // executed if condition 2 is true
        }

        else if (Condition 3) {

        // executed if condition 3 is true

        }
        else {
        // executed if all condition false

        }

//check condition step by step
public class Arithmatic {

        public static void main(String[] args) {

                int marks = 70;
                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) {
                        System.out.println("A grade");
                }else {
                        System.out.println("incorrect input");
                }
        }
}
```

4. Nested if statements

The nested if statement represents the if block within another if block. Here, the inner if block condition executes only when outer if block condition is true.

Syntax

```
if (Condition) {


    if (Condition) {
        }
}
```

```
public class Arithmatic {

    public static void main(String[] args) {

        int no = 75;
        if (no >= 18) {
            if (no > 50) {
                System.out.println("No is greater than 50");
            }
        }
    }
}
```

5. **Switch Statements-**

A switch statement in java is used to execute a single statement from multiple conditions. The switch statement can be used with short, byte, int, long, enum types, etc. Usage of break statement is made to terminate the statement sequence. It is optional to use this statement.

We can use string and int in switch statements.

Syntax

```
switch (expression) {

case 1:
```

```
        Statement 1
        break;

        case 2:
        Statement 2
        break;

        case 3:
        Statement 3
        break;

        default:

        default statement
```

Example-1

```
public class Arithmatic {

        public static void main(String[] args) {

                int number = 2;
                switch (number) {
                case 1:
                        System.out.println("this is 1 number");
                        break;
                case 2:
                        System.out.println("this is 2 number");
                        break;
                case 3:
                        System.out.println("this is 3 number");
                        break;
                default:
                        System.out.println("Invalid input");
                }
        }
}
```

Output-

this is 2 number

Example-2

```
public class SwitchExample {

    public static void main(String[] args) {

        String month = "March";

        switch (month) {
        case "January":
            System.out.println("this is January");
            break;
        case "February":
            System.out.println("this is February");
            break;
        case "March":
            System.out.println("this is March");
            break;
        case "April":
            System.out.println("this is April");
            break;
        case "May":
            System.out.println("this is May");
            break;
        case "June":
            System.out.println("this is June");
            break;
        case "July":
            System.out.println("this is July");
            break;
        case "August":
            System.out.println("this is August");
            break;
        case "September":
            System.out.println("this is September");
            break;
        case "October":
            System.out.println("this is October");
            break;
        case "November":
            System.out.println("this is November");
            break;
        case "December":
            System.out.println("this is December");
            break;
        default:
```

```
                System.out.println("this is Invalid choice...");
            }

        }

}
```

Output

this is March

Example-3

```
public class SwitchExample {

        public static void main(String[] args) {

            Test test = new Test();
            String operation = "Multiplication";
            switch (operation) {
            case "Addition":
                int add = test.getAddition(10, 20);
                System.out.println("Addition is>>" + add);
                break;
            case "Substraction":
                int sub = test.getSubstraction(10, 20);
                System.out.println("Substraction is>>" + sub);
                break;
            case "Multiplication":
                int mul = test.getMultiplication(10, 20);
                System.out.println("Multiplication is>>" + mul);
                break;
            case "Division":
                int div = test. getDivision(10, 20);
                System.out.println("Division is>>" + div);
                break;
            default:
                System.out.println("Incorrect input, Please try
again.");
            }
        }

}
public class Test {

        public int getAddition(int a, int b) {
            int add = a + b;
            return add;
```

```
	}

	public int getSubstraction(int a, int b) {
		int sub = a - b;
		return sub;
	}

	public int getMultiplication(int a, int b) {
		int mul = a * b;
		return mul;
	}

	public int getDivision(int a, int b) {
		int div = a / b;
		return div;
	}

}
```

Output-

Multiplication is>>200