

Classification of numbers

```
class classifyNumbers {  
    public static void main (String arg[]) {  
        int n = 8;  
        switch (n) {  
            case 0 :  
                System.out.println("Zero!");  
                break;  
            case 1 :  
                System.out.println("Smallest positive!");  
                break;  
            case 2 :  
                System.out.println("Smallest prime!");  
                break;  
            // continued in next slide
```

Classification of numbers

case 3 :

```
System.out.println("Smallest odd prime!");  
break;
```

case 4 :

```
System.out.println("Smallest prime squared!");  
break;
```

case 5 :

```
System.out.println("Number of fingers!");  
break;
```

case 6 :

```
System.out.println("Smallest perfect!");  
break;
```

// continued in next slide

Classification of numbers

```
    case 7 :
        System.out.println("North seven stars!");
        break;
    case 8 :
        System.out.println("Smallest prime cubed!");
        break;
    case 9 :
        System.out.println("Smallest odd prime squared!");
        break;
    default :
        System.out.println("Not a digit!");
        break;
} // end of switch
} // end of main
} // end of class
```

More classification

```
class differentClassification {  
    public static void main (String arg[]) {  
        int n = 8;  
        switch (n) {  
            case 2:  
            case 3:  
            case 5:  
            case 7:  
                System.out.println("Prime!");  
                break;  
            case 1:  
            case 4:  
            case 9:  
                System.out.println("Square!");  
                break;  
            // continued in next slide
```

More classification

case 6:

```
    System.out.println("Perfect!");  
    break;
```

case 8:

```
    System.out.println("Cube!");  
    break;
```

case 0:

```
    System.out.println("Zero the Great!");  
    break;
```

default:

```
    System.out.println("Not a digit!");  
    break;
```

```
    } // end switch
```

```
    } // end main
```

```
    } // end class
```

More example of switch

```
class rainbow {  
    public static void main (String arg[]) {  
        char c = 'V';  
        switch (c) {  
            case 'V' :  
            case 'v' :  
                System.out.println ("Violet");  
                break;  
            case 'I' :  
            case 'i' :  
                System.out.println ("Indigo");  
                break;  
            case 'B' :  
            case 'b' :  
                System.out.println ("Blue");  
                break;  
        }  
    }  
}
```

More example of switch

```
case 'G' :  
case 'g' :  
    System.out.println ("Green");  
    break;  
case 'Y':  
case 'y' :  
    System.out.println ("Yellow");  
    break;  
case 'O' :  
case 'o' :  
    System.out.println ("Orange");  
    break;  
case 'R' :  
case 'r' :  
    System.out.println ("Red");  
    break;
```

// continued in next slide

More example of switch

default :

```
        System.out.println ("You are not in  
rainbow!");
```

```
        break;
```

```
    }
```

```
 }
```

```
}
```


Nested loops

- Loop within loop

```
for (i=0; i<=100; i++) {  
    for (j=0; j<=100; j++) {  
        System.out.println (i+j);  
    }  
}
```

- Number of loops is called the depth of the nest
- The innermost loop executes most frequently
- The outermost loop executes least
- You can nest while loops within for loops and vice-versa
- Loop variables should be different for different loops e.g., i and j in this case (what happens if both are i ?)

Nested loops

```
for (i1=p1; i1<q1; i1++) { // outermost
    statements1           // can be empty
    for (i2=p2; i2<q2; i2++) {
        statements2       // can be empty
        for (i3=p3; i3<q3; i3++) {
            statements3    // can be empty
            ...
            for (iN=pN; iN<qN; iN++) { // innermost
                statementsN
            }
            ...
        }
    }
} // How many times does statementsK execute?
```

Nested-for example

```
class tables {  
    public static void main (String args[]) {  
        int i,j,columns,rows;  
        columns=15;rows=15;  
        for (i=1;i<=rows;i=i+1){  
            for (j=1;j<=columns;j=j+1){  
                if (i*j < 10) System.out.print(" "+i*j+" ");  
                else if (i*j < 100) System.out.print(" "+i*j+" ");  
                else System.out.print(i*j+" ");  
            }  
            System.out.println();  
        }  
    }  
}
```

All perfect numbers

```
class allPerfectNumbersUptoOneLakh {
    public static void main (String arg[]) {
        int n, d, sigma_n;
        for (n=2; n<=100000; n++) {
            sigma_n = 1+n;
            for (d=2; d<=n/2; d++) {
                if ((n%d)==0) {
                    sigma_n += d;
                }
            }
            if (sigma_n == 2*n) {
                System.out.println (n + "is perfect.");
            }
        }
    }
}
```