

Useful Segments of JAVA Code

1) Swap two numbers

```
temp = a;
a = b;
b = temp;
```

2) Find the larger of two numbers

```
if (a > b) {
    largest = a;
} else {
    largest = b;
}
OR
largest = a;
if (b > largest){
    largest = b;
}
```

3) Determine total pay with overtime

```
if (hours <= 40) {
    totalpay = hours * payrate;
} else {
    totalpay = 40 * payrate + (hours - 40) * payrate * overtimefactor;
}
```

4) Check for validity of input

```
Scanner reader = new Scanner(System.in);
System.out.print("Enter a nonzero number: ");
double number = reader.nextDouble();

if (number == 0) {
    System.out.println("Error: Number must be nonzero!");
} else {
    double answer;
    answer = 1 / number;
    System.out.println("The reciprocal of your number is " + answer);
}
```

5) Print the numbers from 1 to 100 and their sum. **Count-Controlled loop**

```
int x, sum;
x = 1; sum = 0;

while (x <= 100) {
    System.out.println(x);
    sum = sum + x;
    x = x + 1;
}

System.out.println(sum);
```

6) Adding flexibility to printing the sum and the numbers from start to finish with increment

```
int x, sum, starting, stopping, increment;
starting = 100;  stopping = 300; increment = 10;

x = starting; sum = 0;

while (x <= stopping) {
    System.out.println(x);
    sum = sum + x;
    x = x + increment;
}

System.out.println(sum);
```

7) Counting backward

```
int x;

x = 50;
while (x >= 1) {
    System.out.println(x);
    x = x -1;
}
```

8) Continue to add the numbers 1, 2, 3, ... until the sum has reached 50. Print out the number that causes it to reach a sum of 50. **Task-Controlled loop.**

```
int x, sum;
x = 0;  sum = 0;

while (sum <= 50) {
    x = x +1;
    sum = sum + x;
}

System.out.println(x);
```

9) Continue to ask for input until the desired input is obtained.

```
Scanner reader = new Scanner(System.in);
System.out.print("Enter a positive number: ");
double number = reader.nextDouble();

while (number <= 0) {
    System.out.print("Sorry, you must enter a positive number: ");
    number = reader.nextDouble();
}
```

10) Find the factorial of a number

```
Scanner reader = new Scanner(System.in);
System.out.print("Enter a positive integer: ");
int number = reader.nextInt();

int x, product;

x = 2; product = 1;

while (x <= number) {
    product = product * x;
    x = x + 1;
}

System.out.println("The factorial of " + number + " is " + product);
```

11) Add up a set of given numbers until a value of -9999 is reached. **Sentinel-Controlled loop.**

```
Scanner reader = new Scanner(System.in);
System.out.print("Enter a number (-9999 to quit): ");
double number = reader.nextDouble();

double sum;
sum = 0;

while (number != -9999) {
    sum = sum + number;
    System.out.print("Enter a number (-9999 to quit): ");
    number = reader.nextDouble();
}

System.out.println(sum);
```

12) Print the numbers from 1 to 100 and their sum. **Count-Controlled loop**

```
int sum;
sum = 0;

for (int x = 1; x <= 100; x = x+1 ){
    System.out.println(x);
    sum = sum + x;
}

System.out.println(sum);
```

13) Counting backward

```
for (int x = 50; x >= 1; x = x-1) System.out.println(x);
```

14) Prompting for the length of a list of numbers to add

```
Scanner reader = new Scanner(System.in);
System.out.print("How long is the list? ");
int count = reader.nextInt();

double sum = 0;
double number;

for (int i = 1; i <= count; i = i + 1) {
    System.out.print("Enter a number: ");
    number = reader.nextDouble();
    sum = sum + number;
}

System.out.println(sum);
```

15) Determine all divisors (factors) of an integer

```
Scanner reader = new Scanner(System.in);
System.out.print("Enter a positive integer: ");
int n = reader.nextInt();

for (int x = 1; x <= n; x = x + 1) {
    if (n % x == 0) {
        System.out.println(x);
    }
}
```

16) A better method for finding the factors of an integer

```
Scanner reader = new Scanner(System.in);
System.out.print("Enter a positive integer: ");
int n = reader.nextInt();
int limit = Math.sqrt(n);

for (int x = 1; x <= limit; x = x + 1) {
    if (n % x == 0) {
        System.out.println(x);
        System.out.println(n / x);
    }
}
```

17) Is a Number Prime? Method #1

```
Scanner reader = new Scanner(System.in);
System.out.print("Enter a positive integer: ");
int n = reader.nextInt();
int count = 0;

for (int x = 1; x <= n; x = x + 1) {
    if (n % x == 0) {
        count = count + 1;
    }
}

if (count == 2) {
    System.out.println(n + " is a prime number.");
} else {
    System.out.println(n + " is a composite number.");
}
```

18) Is a Number Prime? Method #2

```
Scanner reader = new Scanner(System.in);
System.out.print("Enter a positive integer: ");
int n = reader.nextInt();

boolean prime = true;

for (int x = 2; x <= Math.sqrt(n); x = x + 1) {
    if (n % x == 0) {
        prime = false;
        break;
    }
}

if (prime) {
    System.out.print(n + " is a prime number.");
} else {
    System.out.print(n + " is a composite number.");
}
```

19) Breaking out of infinite loops

```
Scanner reader = new Scanner(System.in);
int number;

While (true) {
    System.out.print("Enter an integer :");
    number = reader.nextInt();
    if (number == -9999){
        break;
    }
    System.out.println(number);
}
```

20) Loop to ask if you wish to do another calculation using do...while

```
Scanner reader = new Scanner(System.in);
double number, answer;
String doanother;

do {
    System.out.print("Enter a number: ");
    number = reader.nextDouble();

    answer = Math.pow(number, 2);
    System.out.println("The square of " + number + " is " + answer);

    System.out.print("Do you wish to do another calculation (Y/N)? ");
    doanother = reader.nextLine();
    doanother = reader.nextLine();

} while (doanother.equals("Y"));
```

21) Reading numbers from a text file until EOF

```
Scanner reader = new Scanner(new File("numbers.txt"));
double number, sum = 0;
int x = 0;

while (reader.hasNext()) {
    number = reader.nextDouble();
    sum = sum + number;
    x = x + 1;
}
System.out.println("The " + x + " numbers have a sum of " + sum);
```

22) Files and Exceptions → Find the average of the numbers found in a text file.

```
import java.io.*;
import java.util.Scanner;

public class ComputeAverage {
    public static void main(String[] args) throws IOException{

        Scanner reader = new Scanner(new File("numbers.txt"));
        double number, sum = 0;
        int x = 0;

        while (reader.hasNext()) {
            number = reader.nextDouble();
            sum = sum + number;
            x = x + 1;
        }

        if (x == 0) {
            System.out.println("The file had no numbers.");
        } else {
            System.out.println("The average is " + sum/x);
        }
    }
}
```

23) Reading and Writing to Text Files

```
import java.io.*;
import java.util.Scanner;

public class FilterEvenOdd {
    public static void main(String[] args) throws IOException{

        Scanner reader = new Scanner(new File("integers.txt"));
        PrintWriter writer1 = new PrintWriter(new File("odd.txt"));
        PrintWriter writer2 = new PrintWriter(new File("even.txt"));
        int number;
        while (reader.hasNext()) {
            number = reader.nextInt();

            if (number % 2 == 0) {
                writer2.println(number);
            } else {
                writer1.println(number);
            }
        }

        writer1.close();
        writer2.close();
    }
}
```

24) Initialization Error → Find sum of odd numbers between 1 and 100.

```
int x = 1;
int sum;
                \\ int sum = 0;

while (x <= 100) {
    sum = sum + x;
    x = x + 2;
}
System.out.println(sum);
```

25) Off-by-One Error → Find sum of odd numbers between 1 and 100.

```
int x = 1;
int sum = 0;

while (x < 99) {
    sum = sum + x;
    x = x + 2;
}
System.out.println(sum);
                \\ while (x<=99) {
```

26) Infinite Loop Error → Find sum of odd numbers between 1 and 100.

```
int x = 1;
int sum = 0;

while (x != 100) {
    sum = sum + x;
    x = x + 2;
}
System.out.println(sum);
                \\ while (x<=100) {
```

27) Error in Loop Body → Find sum of odd numbers between 1 and 100.

```
int x = 1;
int sum = 0;

while (x <= 100) {
    sum = sum * x;
    x = x + 2;
}
System.out.println(sum);
                \\ sum = sum + x;
```

28) Update Error → Find sum of odd numbers between 1 and 100.

```
int x = 1;
int sum = 0;

while (x <= 100) {
    x = x + 2;           // sum = sum + x;
    sum = sum + x;      // x = x + 2;
}
System.out.println(sum);
```

29) Limited Floating-Point Precision

```
double x;
for (x = 0.0; x != 1.1; x = x + 0.1) {
    System.out.print(x + " ");
}
```

Expected Output

0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

Actual Output

0.1 0.2 0.30000000000000004 0.4 0.5 0.6 0.7 0.7999999999999999
0.8999999999999999 0.9999999999999999 1.0999999999999999 1.2 1.3

Corrected Program

```
double x, delta;
delta = 0.0001;

for (x = 0.0; x <= 1.0 + delta; x = x + 0.1) {
    System.out.print(x + " ");
}
```

- OR -

```
for (int x = 0; x <= 10; x = x + 1){
    System.out.print(x / 10.0 + " ");
}
```