



# Examples



Week 6

# ForSample.java

---

```
public class ForSample
{
    public static void main(String[] args)
    {
        int sum=0;

        for (int i=1; i<=10; i++) {
            sum += i;
            System.out.print(i);

            if (i<=9)
                System.out.print("+");
            else {
                System.out.print("=");
                System.out.println(sum);
            }
        }
    }
}
```

1+2+3+4+5+6+7+8+9+10=55

# WhileSample.java

---

```
import java.util.Scanner;

public class WhileSample
{
    public static void main(String[] args)
    {
        int count=0;
        int sum=0;
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter numbers (-1 denotes the end of input) ");

        int n = scanner.nextInt();
        while(n != -1) {
            sum += n;
            count++;
            n = scanner.nextInt();
        }
    }
}
```

# WhileSample.java

---

```
        if(count == 0)
            System.out.println("No input");
        else {
            System.out.println("Count: " + count);
            System.out.println("Average: " + (double)sum/count);
        }
        scanner.close();
    }
}
```

10 30 -20 40 -1 ↵  
Count: 4  
Average: 15.0

# DoWhileSample.java

---

```
public class DoWhileSample
{
    public static void main (String[] args)
    {
        char c = 'a';

        do {
            System.out.print(c);
            c = (char) (c + 1);
        } while (c <= 'z');
    }
}
```

abcdefghijklmnopqrstuvwxyz

# NestedLoop.java

---

```
public class NestedLoop
{
    public static void main(String[] args)
    {
        for(int i=1; i<10; i++)
        {
            for(int j=1; j<10; j++)
            {
                System.out.print(i + "*" + j + "=" + i*j);
                System.out.print('\t');
            }
            System.out.println();
        }
    }
}
```

# NestedLoop.java

---

1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32	4*9=36
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

# ContinueExample.java

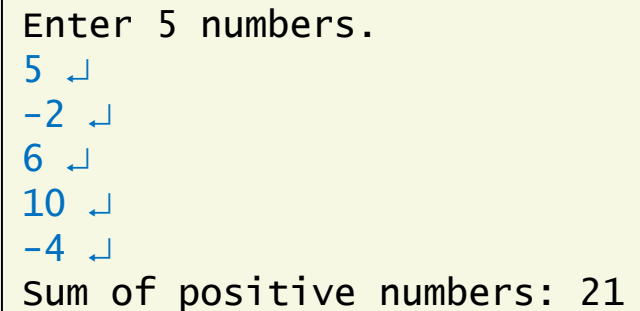
---

```
import java.util.Scanner;

public class ContinueExample {
    public static void main( String[] args ) {
        Scanner scanner = new Scanner( System.in );

        System.out.println( "Enter 5 numbers." );
        int sum=0;
        for(int i=0; i < 5; i++) {
            int n = scanner.nextInt();
            if( n <= 0 )
                continue;
            else
                sum += n;
        }
        System.out.println( "Sum of positive numbers: " + sum );

        scanner.close();
    }
}
```



Enter 5 numbers.  
5 ↵  
-2 ↵  
6 ↵  
10 ↵  
-4 ↵  
Sum of positive numbers: 21



# BreakExample.java

```
import java.util.Scanner;

public class BreakExample
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Type exit to terminate.");
        while(true) {
            System.out.print(">> ");
            String text = scanner.nextLine();
            if(text.equals("exit"))
                break;
        }
        System.out.println("Terminating...");
        scanner.close();
    }
}
```

```
Type exit to terminate.
>> edit ↵
>> babo ↵
>> Exit ↵
>> exit ↵
Terminating...
```

# ArrayAccess.java

```
import java.util.Scanner;

public class ArrayAccess
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        int intArray[] = new int[5];

        int max=0;
        System.out.println("Enter 5 positive integers.");
        for (int i=0; i<5; i++) {
            intArray[i] = scanner.nextInt();
            if (intArray[i] > max)
                max = intArray[i];
        }
        System.out.println( "Max: " + max );
        scanner.close();
    }
}
```

Enter 5 positive integers.

1 ↵

39 ↵

78 ↵

100 ↵

99 ↵

Max: 100

# Ex06\_1.java

---

- ▶ 최대값, 최소값, 합계, 평균, 분산, 표준편차
- ▶ 아래 배열에 저장된 점수(0~100점)의 최대값, 최소값, 평균, 분산, 표준편차를 계산하여 출력하시오.
  - ▶ `int[] x = { 87, 68, 94, 100, 83, 78, 85, 91, 76, 87, 55, 60, 99, 63, 80 };`

$$m = \frac{x_1 + x_2 + \cdots + x_N}{N}$$

$$V = \frac{(x_1 - m)^2 + (x_2 - m)^2 + \cdots + (x_N - m)^2}{N}$$

$$\sigma = \sqrt{V}$$

```
max = 100
min = 55
m = 80.40
V = 179.04
sigma = 13.38
```

✓ 참고

`Math.pow( x, y )`:  $x^y$  을 리턴

`Math.sqrt( x )`:  $\sqrt{x}$  ( $x$ 의 제곱근)을 리턴)

# Ex06\_2.java

---

- ▶ 복리이자 응용
- ▶ 어떤 사람이 \$1000를 연이자 5%의 정기예금에 들었다. 원리합계가 매년 다시 투자된다고 할 때, 10년 동안의 매년 연말 원리합계를 계산하여 출력하라. 다음 공식을 이용하시오.

$$a = p (1 + r)^n$$

where

$p$ : 원금

$r$ : 이자율 (5%일 때  $r=0.05$ )

$n$ : 기간

$a$ :  $n$ 번째 해 연말 원리합계

- ▶ `Math.pow( x, y )`:  $x^y$  을 리턴

Year	Amount on deposit
1	1,050.00
2	1,102.50
3	1,157.63
4	1,215.51
5	1,276.28
6	1,340.10
7	1,407.10
8	1,477.46
9	1,551.33
10	1,628.89

# Ex06\_3.java

---

- ▶ 설문조사
- ▶ 학생식당에 대한 만족도 설문조사의 결과가 아래와 같이 배열에 저장되어 있다. 만족도 점수는 1~10점이다. 각 점수별 응답자의 수를 세어 출력하시오.

```
int[] responses = { 1, 2, 6, 4, 8, 5, 9, 7, 8, 10,  
                    1, 6, 3, 8, 6, 10, 3, 8, 2, 7,  
                    6, 5, 7, 6, 8, 7, 5, 6, 6, 5,  
                    6, 7, 5, 6, 4, 8, 6, 8, 10, 6};
```

Rating	Frequency
1	2
2	2
3	2
4	2
5	5
6	11
7	5
8	7
9	1
10	3

# Ex06\_4.java

---

- ▶ 구구단
- ▶ 2~9단을 한 단씩 출력하시오.
- ▶ 아래와 같은 형식으로 출력하시오.

```
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
```

```
3 x 1 = 3
3 x 2 = 6
...
```



# Ex06\_5.java

---

- ▶ 구구단
- ▶ 2~9단을 두 단씩 출력하시오.
- ▶ 아래와 같은 형식으로 출력하시오.

2 x 1 = 2	3 x 1 = 3
2 x 2 = 4	3 x 2 = 6
2 x 3 = 6	3 x 3 = 9
2 x 4 = 8	3 x 4 = 12
2 x 5 = 10	3 x 5 = 15
2 x 6 = 12	3 x 6 = 18
2 x 7 = 14	3 x 7 = 21
2 x 8 = 16	3 x 8 = 24
2 x 9 = 18	3 x 9 = 27
4 x 1 = 4	5 x 1 = 5
4 x 2 = 8	5 x 2 = 10
...	...



# Ex06\_6.java

---

- ▶ 짝수 패리티(even parity)
- ▶ 8 개의 0과 1로 이루어진 데이터 비트가 주어질 때, 패리티 비트를 맨 마지막에 추가한 9 개 데이터 비트를 출력하시오.

0 1 1 0 1 0 0 1 ↵
0 1 1 0 1 0 0 1 0

