

## Control flow

→ if-else

if (condition) {

// runs if condition is

true

else {

// runs if condition is

false

↳ import java.util.Scanner;

eg: class classA {

public static void main (String[] args) {

Scanner sc = new Scanner (System.in);

int marks1 = sc.nextInt();

int marks2 = sc.nextInt();

if (marks1 == 33 & marks2 == 33) {

System.out.println ("Pass!");

else {

System.out.println ("Fail!");

}

100 100 100

Eg: int marks = sc.nextInt();

if (marks == 33) {

System.out.println ("Pass!");

else {

System.out.println ("Fail!");

}

Instead of if-else we can use  
ternary operator

condition ? true : false

Eg: int marks = sc.nextInt();  
boolean passed = marks >= 33 ? true : false;  
char grade = marks >= 33 ? 'P' : 'F';

if, ... else if, ... else

(if else ladder)

Eg: if (marks >= 80)  
    v. Good;  
else if (marks >= 50)  
    Good;  
else if (marks >= 33)  
    Pass;  
else  
    Fail;

### Problems

1) Given 2 integers, print the one that has  
larger value.

Approach:

- ① a, b
- if  $a > b \Rightarrow$  print a
- if  $a < b \Rightarrow$  print b
- if  $a = b \Rightarrow$  print any of the value

Code:

```
class Learnyard {
    public static void main (String [] args) {
```

```
        Scanner sc = new Scanner (System. in);
```

```
        int a = sc.nextInt();
```

```
        int b = sc.nextInt();
```

```
        if (a > b) {
```

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

```
}
```

```
        else if (a < b) {
```

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

```
}
```

```
        else {System.out.println ("a");}
```

### Approaches

- Brute
- Better
- Optimal

Calculate Time Complexity  
& Space Complexity

o) max of three integers

Eg: class className

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int a = sc.nextInt();
    int b = sc.nextInt();
    int c = sc.nextInt();
    if (a >= b && a >= c)
        System.out.println(a);
    else if (b >= a && b >= c)
        System.out.println(b);
    else
        System.out.println(c);
```

Go to Codeforces  $\Rightarrow$  Watermelon problem

watermelon has  $w$  kilos

I/P: 8

O/P: Yes

If it can divide into 2 parts with even kilos  
then print Yes else No

Eg: I/P: 8  $\Rightarrow$  (4,4), (6,2)

O/P: Yes

2) I/P: 7  $\Rightarrow$  (5,2)  $\rightarrow$  odd

O/P: No

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int w = sc.nextInt();
```

if ( $w > 2$  &&  $w \% 2 == 0$ )
 System.out.println("Yes");

else

```
System.out.println("No");
```

g

Q1 take an ~~int~~ integer i/p from user().

Print all these numbers from 1to N in increasing order

e.g: I/P  $\Rightarrow$  N = 3 O/P  $\Rightarrow$

1  
2  
3

```
for(int i=0; i<=n; i++) {  
    cout << "Practice hard!";  
}
```

I/P  $\Rightarrow$  N = 5

0 / P  $\Rightarrow$

5  
4  
3  
2  
1

```
int n = segmentInt();  
for( int i=n; i>=1; i-- ) {
```

cout << i;

Q1 Take an int & print all the  
eg:  $\text{int } N = 3$

### Q1/P:

- 1 → Monday
- 2 → Tuesday
- 3 → Wednesday
- 4 → Thursday
- 5 → Friday
- 6 → Saturday
- 7 → Sunday

Eg:

```
public static void main
    (String[] args) {
    Scanner sc = new Scanner
        (System.in);

    int day_num = sc.nextInt();
    switch(day_num) {
        case 1:
            System.out.println("Monday");
            break;
        Case 2:
            System.out.println("Tuesday");
            break;
        Case 3:
            System.out.println("Wednesday");
            break;
        Case 4:
            System.out.println("Thursday");
            break;
        Case 5:
            System.out.println("Friday");
            break;
        case 6:
            System.out.println("Saturday");
            break;
        default:
            System.out.println("Sunday");
    }
}
```

If there are multiple statements  
conditions to execute it is  
better to use switch statement.

### Switch Syntax

switch (condition) {

case 1:

    // statements

    break;

case 2:

    // statements

    break;

....

default:

    // statements

    break;

}

### Loops

types of loops ⇒

1) For loop

2) While loop

3) Do-while loop

→ loops are used to repeatedly execute a block  
of code until a certain condition is met.

1) for loop

```
for(initialization; condition; update) {
    code to be executed
}
```

Eg: `for (int ctr=0; ctr<=20; ctr++) {  
 System.out.println("Practice!");  
}`