

3.1 : Boolean Expressions

• Testing Equality (`==`, `!=`) :

`==` is used to check if two values are equal.

`!=` is used to check if two values are not equal.

Example:

```
public static void main (String[] args) {  
    int x = 54;  
    int y = 82;  
    int z = 54;  
    System.out.println (x==y); ← prints false  
    System.out.println (x==z); ← prints true  
    System.out.println (x!=y); ← prints true  
}
```

comparing primitive data types

Example :

```
public class Student {  
    String name;  
    int age;  
  
    Student () {  
        name = "Sam";  
        age = 15  
    }  
}
```

```
public class School {
```

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

```
        Student std1 = new Student(); ← creating a new Student object
```

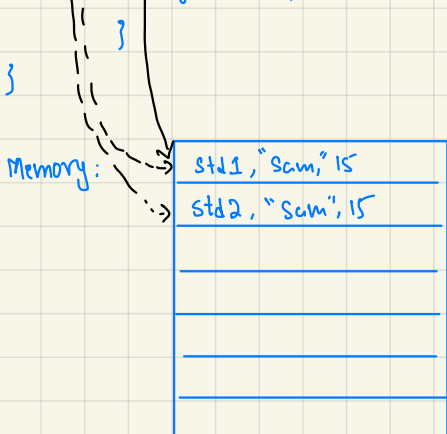
```
        Student std2 = new Student(); ← creating a new Student object
```

```
        Student a = std1; ← setting a reference variable "a" that is set to std1.
```

```
        System.out.println (std1==std2); ← prints false (comparing 2 different objects)
```

```
        System.out.println (std1.age==std2.age); ← prints true (comparing 2 primitive values)
```

```
        System.out.println (std1==a); ← prints true (they both are referencing to the same object)
```



These two objects are different.

It doesn't matter if the values are same for two different objects.

Note: We use compareTo and equals operator to compare two strings.

Java compiler will warn us if we use `"=="` or `"!="`.

```
String s1 = "hello";  
String s2 = "hello";
```

s1 and s2 are two different objects so `==`, `!=` cannot be used.

String is class type, not primitive.

• Relational Operators (<, >)

- In Java, these are only used to compare numeric values or arithmetic expressions.
- Use compareTo, equals for comparing String values.

<, >, <=, >=, ==, !=

Example: public static void main (String[] args) {

int a = 4;

int b = 8;

System.out.println(a > b);

System.out.println(b < a);

System.out.println(b <= a * 2);

System.out.println(a >= b / 2);

}

• Remainder (%)

Ex: public static void main (String[] args) {

int a = 5;

int b = 6;

System.out.println("It is " + (a % 2 == 0) + "that " + a + "is even");

System.out.println("It is " + (b % 2 == 0) + "that " + b + "is even");

}

checking if number is even.

Note: You can also use remainder to check if a number is divisible by another number. (num1 % num2 == 0)

8 % 2 == 0
0 == 0 (True)

Use to get the last digit from an integer number: (num % 10)

505 % 10

5

Ex: Convert number of minutes to hrs and minutes

int totalmins = 125;

int hrs = totalmins / 60; ← 2

int mins = totalmins % 60; ← 5

2hrs and 5mins.

Note: Use num % 2 != 0 to check if a number is odd.

If you do num % 2 == 1, and you use a negative odd number, it will return false

-7 % 2

-1 -1 is not equal to 1 (it would return false) X

Ex: Simple If statements.

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

```
    int a = 5;
```

```
    int b = 10;
```

```
    String str1 = "hello";
```

```
    String str2 = "world";
```

if (a > b) { ← only runs if this statement evaluates to true.

```
    System.out.println("variable a is greater than variable b");
}
```

if (!str1.equals(str2)) { ← only runs if this statement evaluates to true

```
    System.out.println(str1 + " is not equal to " + str2);
}
```

```
}
```

Note: Don't put semi colons at the end of if statements. Use curly braces to enclose block of statements under the if condition

Ex2: public static void main (String[] args) {
 boolean isMonday = true;

```
    if (isMonday) {
```

```
        System.out.println("Happy Monday");
```

```
    }
```

```
    System.out.println("Have a great day"); ← This line would print regardless of the value of isMonday
```

If isMonday is true (in example)

output: Happy Monday

Have a great day

If isMonday is false:

output: Have a great day

Ex3: public static void main (String[] args) {
 int number = (int) (Math.random() * 11) - 5;

```
    if (number > 0) {
```

```
        System.out.println(number + " is Positive");
```

```
    }
```

```
    if (number == 0) {
```

```
        System.out.println(number + " is Zero");
```

```
    }
```

```
    if (number < 0) {
```

```
        System.out.println(number + " is negative");
```

```
    }
```

```
}
```

3.3 If - else statements

Example: `public static void main (String[] args) {`

`int a = 4;
int b = 5;`

`if (a > b) {`

`System.out.println(a + " is greater than " + b);`

`}`

`else {`

`System.out.println(b + " is greater than " + a);` ← should print this line

`}`

If we have only one statement in if/else, curly braces are optional

ex: `if (a < b)
System.out.println(...)
else
System.out.println(...)`

Example:

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

`boolean isMonday = false;`

`if (isMonday) {`

`System.out.println("Send zoom link to Ali");`

`} else {`

`System.out.println("Do not send zoom link to Ali");`

`}`

`System.out.println("Practice questions and read notes");`

`}`

changing this to true will change the output. It will go to the if block and skip else block.

This code should print:

Do not send zoom link to Ali
Practice questions and read notes

Nested Ifs and Dangling Else.

- It is possible to have if statements inside other if statements (Nested Ifs).
- Dangling else: With multiple ifs (nested ifs), we can find a else (dangling else) that can belong to either if statement.

Note: The else clause will always be part of the closest unmatched if statement in the same block of code, regardless of indentation.

Example:

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

`int a = 4;`

`int b = 5;`

`int c = 7;`

`if (a > b)`

`{ if (b > c)`

`System.out.println(b + " is greater than " + c + " but less than " + a);`

`else` ← belongs to the nearest if (b > c)

`System.out.println(b + " is less than " + a + " and " + c);`

`} else`

`System.out.println(b + " is greater than " + a);`

`}`

Indentation doesn't matter. It is done to make it clear

← $(1 > 5)$ is false so it would go to the corresponding else block

This block will only execute if $a > b$

This else belongs to $a > b$

Output: 5 is greater than 1.

Example 2: `public static void main (String[] args) {`

`int a=1;
int b=2;
int c=3`

`if (a > b)`

← This will not print anything since there is no else block for this if.

`if (a > c)`

This else belongs to `if (a > c)`

`else`

`System.out.println (a + " is greater than " + b + " and " + c);`

`System.out.println (a + " is greater than " + b + " but not " + c);`

Practice Question 1: `public static void main {`

`int a = 1;
int b = 2;
int c = 3;`

`if (a > b)`

`if (b > c)`

`System.out.println (b + " is greater than " + c + " but less than " + a);`

`else`

`System.out.println (b + " is less than " + a + " and " + c);`

`else`

`System.out.println (b + " is greater than " + a);`

`}`

What will be the values when :

(i) `a=1, b=2, c=3`

(ii) `a=3, b=2, c=1`

(iii) `a=3, b=1, c=2`

(iv) `a=3, b=3, c=2`

(v) `a=3, b=2, c=3`

(vi) `a=2, b=3, c=3`

Practice Question 2:

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

`int x = 1;`

`int y = 2;`

`int z = 3;`

What will be the output when:

(i) `x=1, y=2, z=3`

(ii) `x=3, y=2, z=1`

(iii) `x=3, y=1, z=4`

`if (x > y)`

`if (x > z)`

`System.out.println (x + " is greater than " + y + " and " + z);`

`else`

`System.out.print (x + " is greater than " + y + " but not " + z);`

`}`

Note: To specify the else clause to be part of top if, we can use curly braces.

```
Ex: public static void main (String[] args) {  
    int a=5;  
    int b=6;  
    int c=7;  
  
    if (a > b) {  
        if (a > c)  
            System.out.println(a + " is greater than " + b + " and " + c);  
    }  
    else  
        System.out.println(a + " is not greater than both " + b + " and " + c);  
}
```

use of curly braces to define if block

top if block ends.

This else now belongs to top if.

Practice Question 3:

```
public static void main (String[] args) {  
    int x=1;  
    int y=2;  
    int z=3;  
  
    if (x > y) {  
        if (x > z)  
            System.out.println(x + " is greater than " + y + " and " + z);  
    }  
    else  
        System.out.print(x + " is not greater than " + y);  
}
```

What will be the output when:

- (i) x=1, y=2, z=3
- (ii) x=3, y=2, z=1
- (iii) x=3, y=1, z=4

3.4 if, else-if, else :

Example:

```
public static void main (String[] args) {  
    int x=2;  
    if (x < 0) {  
        System.out.println("x is negative");  
    }  
    else if (x == 0) {  
        System.out.println("x is zero");  
    }  
    else {  
        System.out.println("x is positive");  
    }  
    System.out.println("Happy Tuesday");  
}
```

← skips this if block when x=2

← skips this block when x=2

← enters the else block and prints: x is positive

← This is not part of any if block, it will execute after if, else-if, else.

Output: x is positive
Happy Tuesday.

Practice Question 4:

```
public static void main (String[] args) {  
    if (x < 0.25) {  
        System.out.println("first quartile");  
    }  
    else if (x < 0.5) {  
        System.out.println("second quartile");  
    }  
    else if (x < 0.75) {  
        System.out.println("third quartile");  
    }  
    else {  
        System.out.println("fourth quartile");  
    }  
}
```

What will be the output when: (assume x is a double)

(i) $x = 0.5$

(ii) $x = 0.72$

(iii) $x = 22$

Practice Question 5:

```
public static void main (String[] args) {  
    int score = 56;  
    String grade = "";  
  
    if ((score >= 90) && (score <= 100)) {  
        grade = "NICK'S GRADE";  
    }  
    else if (score >= 80) {  
        grade = "B";  
    }  
    else if (score >= 70) {  
        grade = "C";  
    }  
    else if (score >= 60) {  
        grade = "D";  
    }  
    else if (score >= 0) {  
        grade = "F";  
    }  
    else {  
        grade = "INVALID";  
    }  
    System.out.println(grade);  
}
```

what will be the output when:

(i) Score = 54

(ii) Score = 84

(iii) Score = 99

(iv) Score = -5

(v) Score = 125