

# Java

## Lesson 2

# Summary

- Comparison Operators
- Logical Operators
- Conditional Statements
- Loops

# Comparison Operators

# Comparison Operators in Java

Name of the Operator	Operator	Example
Equal to	<code>==</code>	<code>a == b</code>
Not equal to	<code>!=</code>	<code>a != b</code>
Less than	<code>&lt;</code>	<code>a &lt; b</code>
Greater than	<code>&gt;</code>	<code>a &gt; b</code>
Less than or equal to	<code>&lt;=</code>	<code>a &lt;= b</code>
Greater than or equal to	<code>&gt;=</code>	<code>a &gt;= b</code>



# Logical Operators

## Logical Operators in Java

### 1. Logical AND Operator (& and &&)

Operand1	Operand2	Returned Value
False	False	False
False	True	False
True	False	False
True	True	True

### 2. Logical OR Operator (| and ||)

Operand1	Operand2	Returned Value
False	False	False
False	True	True
True	False	True
True	True	True

### 3. Logical NOT Operator (!)

Operand	Returned Value
False	True
True	False

## Logical Operators in Java

### 4. Logical XOR Operator (^)

Operand1	Operand2	Returned Value
False	False	False
False	True	True
True	False	True
True	True	False



If statements

- If statements allow us to build programs that can make decisions based in certain conditions

```
int temp=32;  
if (temp>30){  
    System.out.println("hot day");  
}  
else if (temp>20)  
    System.out.println("Beautiful day");  
else  
    System.out.println("Cold day");
```



# Simplify if statements

```
int income=2000;  
boolean hasHighIncome=false;  
If (income>1500){  
    hasHighIncome=true;  
}
```

**We can simplify it to**

```
int income=2000;  
boolean hasHighIncome=income>1500 ;
```



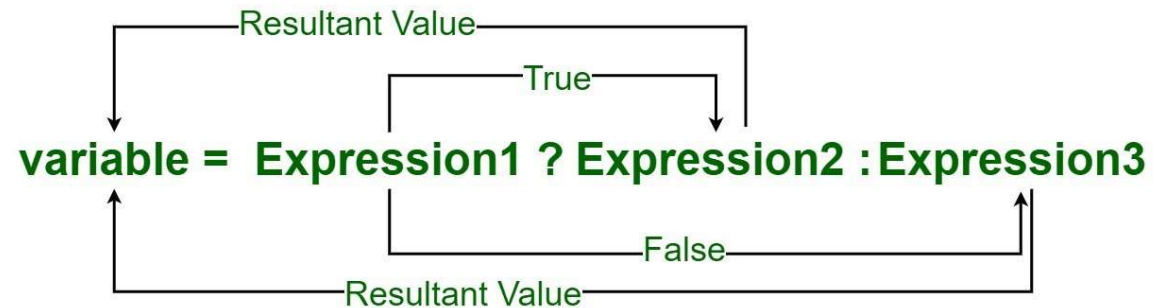
Ternary operator

# If-then-else in one line

---

```
if(Expression1)
{
    variable = Expression2;
}
else
{
    variable = Expression3;
}
```

## Conditional or Ternary Operator (?:) in Java



# Example

```
//choose the className depending on the passengers income  
int income=120_000;  
String className=income>100_000; "First_class": "Economy"
```



Switch

- Instead of writing many if..else statements, you can use the switch statement.
- The switch statement selects one of many code blocks to be executed:
- Note that if the default statement is used as the last statement in a switch block, it does not need a break.

```
switch(expression) {  
  case x:  
    // code block  
    break;  
  case y:  
    // code block  
    break;  
  default:  
    // code block  
}
```

fizzBuzz

- Write a program that behaves like this

- You enter a Number, if this number is divisible by five, we get a Fizz
- if this number is divisible by three, we get a Buzz
- if this number is divisible by both five and three, we get a FizzBuzz
- if this number is not divisible by five or three, we get the number

- We need to know how to read an object

- If it is a string, you should make it integer



# is this correct?

```
import java.util.Scanner; // import the Scanner class

class Main {

    public static void main(String[] args) {

        Scanner myObj = new Scanner(System.in);

        String numberStr;

        System.out.println("Enter number ");

        numberStr = myObj.nextLine(); // this is a String

        int number = Integer.parseInt(numberStr);

        if (number%5==0){

            System.out.println("Fizz");

        }else if (number%3==0){

            System.out.println("Buzz");

        }else if (number%5==0&&number%3==0){

            System.out.println(" FizzBuzz ");

        }

        else

            System.out.println("Number"+ number);

    }

}
```

**\*\* try to insert 5, 3, 15 what result do you get?**

- We have to place the most specific conditions at the top

```
import java.util.Scanner; // import the Scanner class
class Main {
    public static void main(String[] args) {
        Scanner myObj = new Scanner(System.in);
        String numberStr;
        System.out.println("Enter number ");
        numberStr = myObj.nextLine(); // this is a String
        int number = Integer.parseInt(numberStr);
        if (number%5==0&&number%3==0){
            System.out.println("FizzBuzz");
        }else if (number%3==0){
            System.out.println("Buzz");
        }else if (number%5==0){
            System.out.println("Fizz");
        }
        else
            System.out.println("Number"+ number);
    }
}
```



Loops

- **For Loop**

```
for (int i=0;i<5;i++){
```

```
}
```

```
for (int i=5;i>0;i--){
```

```
}
```

- If we don't know how many times we execute a program we use while loops

- **While Loops**

```
String input=" ";
```

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

```
while (!input.equals("quit")){
```

```
    System.out.println("Input ");
```

```
    input=scanner.next().toLowerCase();
```

```
}
```

- **Do...while**

- It is executed at least once

```
String input=" ";
Scanner scanner=new Scanner(System.in);
do {
    System.out.println("Input ");
    input=scanner.next().toLowerCase();
} while (!input.equals("quit"));
```

- **Break and Continue**

```
String input=" ";
Scanner scanner=new Scanner(System.in);
while (!input.equals("quit")){
    System.out.println("Input ");
    input=scanner.next().toLowerCase();
    If (input.equals("pass"))
        continue;//moves control to
the beginning of the loop
    If (input.equals("quit"))
        break;// termination of the
loop
    System.out.println(input );
}
```

```
While (true){
```

```
}// you pass true in the while  
loop, it will be infinitive while  
loop.
```

- To exit the loop you have to use **break**

```
while (true) {  
    ....  
    if (obj == null) {  
        break;  
    }  
    ....  
}
```

For-each is another **array** traversing technique like for loop, while loop, do-while loop introduced in Java

```
for (type var : array)
```

```
{
```

```
    statements using var;
```

```
}
```



# Exercise Mortgage Calculator



For each question we want to validate the value that the user enters, if the value is invalid, we want to keep asking the same question

we can use an **infinite loop**



```
1. import java.util.Scanner;
2. import java.text.NumberFormat;
3. class MortgageCalulator {
4.     public static void main(String[] args) {
5.         final byte MONTHS_IN_YEAR=12;
6.         final byte PERCENT=100;
7.         Scanner scanner=new Scanner(System.in);
8.         System.out.print("Principal:");
9.         int principal=scanner.nextInt();
10.        System.out.print("Annual interest rate
        (epitokio):");
```

```
11.     float interestRate=scanner.nextFloat();
12.         float
        monthlyInterestRate=interestRate/PERCENT/MONTHS_IN_YEAR;
13.         System.out.print("Number of years:");
14.         byte years=scanner.nextByte();
15.         int numberOfPayments=years*MONTHS_IN_YEAR;
16.         double mortgage=principal*monthlyInterestRate*
        Math.pow(1+monthlyInterestRate,numberOfPayments)/(Ma
        th.pow(1+monthlyInterestRate,numberOfPayments)-1);
17.         System.out.println("Mortage: "+mortgage);
18.         String
        mortgageFormatted=NumberFormat.getCurrencyInstance().format(
        mortgage);
19.         System.out.println("Mortage: "+mortageFormatted);
20.     }
21. }
```

### At line 8 we have our first question

```
System.out.print("Principal:");
```

```
int principal=scanner.nextInt();
```

We can add an **infinite loop** until the user enters a valid value ( $1000 < \text{principal} < 1000000$ )

```
while (true) {  
    System.out.print("Principal:");  
    int principal =scanner.nextInt();  
    if (principal >=1000 && principal <=1000000)  
        break;  
    System.out.println("Enter a value between  
[1000,1000000]");  
}
```

If you add these lines in your code you will get an error in **principal**

Why?

- When you declare a variable inside a block, that variable is scoped in the block in which it's defined, and it is not available outside this block.
- To solve the problem, we have to declare it outside the while loop

```
int principal=0;  
while (true) {  
    System.out.print("Principal:");  
    principal =scanner.nextInt();  
    if (principal >=1000 && principal <=1000000)  
        break;  
    System.out.println("Enter a value between  
[1000,1000000]");  
}
```

# Repeat the same pattern for all questions

```
1. import java.util.Scanner;
2. import java.text.NumberFormat;
3. class MortgageCalulator {
4.     public static void main(String[] args) {
5.         final byte MONTHS_IN_YEAR=12;
6.         final byte PERCENT=100;
7.         int principal=0;
8.         float interestRate=0;
9.         byte years=0;
10.        int numberOfPayments=0;
11.        Scanner scanner=new Scanner(System.in);
12.        while (true) {
13.            System.out.print("Principal:");
14.            principal =scanner.nextInt();
15.            if (principal >=1000 && principal <=1000000)
16.                break;
17.            System.out.println("Enter a value between
[1000,1000000]");
}
```

```
while (true) {
    System.out.print("Annual interest rate (epitokio):");
    interestRate=scanner.nextFloat();
    If (interestRate>=1 && interestRate<=30)
        break;
    System.out.println("Enter a value between [1,30]");
}
float monthlyInterestRate=interestRate/PERCENT/MONTHS_IN_YEAR;
while (true) {
    System.out.print("Number of years:");
    years=scanner.nextByte();
    If (years >=1 && years <=30){
        numberOfPayments=years*MONTHS_IN_YEAR;
        break;
    }
    System.out.println("Enter a value between [1,30]");
}
double mortgage=principal*monthlyInterestRate*
    Math.pow(1+monthlyInterestRate,numberOfPayments)/(Math.pow(1+monthlyInterestRate,num
berOfPayments)-1);
System.out.println("Mortage: "+mortage);
String mortgageFormatted=NumberFormat.getCurrencyInstance().format(mortage);
System.out.println("Mortage: "+mortageFormatted);
}
```



Can we make it cleaner?

- To enter a new value between a range we will create a method

- `public static double readNumber(String print_text, double min, double max)`

- To calculate the mortgage

- `public static double calculateMortgage(int principal, float annualInterest, byte years){`

- To print the mortgage we will create a method

- `printMortgage(int principal, float annualInterest, byte years)`

```
public static double readNumber(String printText,
double min, double max) {
    Scanner scanner = new Scanner(System.in);
    double value;
    while (true) {
        System.out.print(printText);
        value = scanner.nextFloat();
        if (value >= min && value <= max)
            break;
        System.out.println("Enter a value between " +
min + " and " + max);
    }
    return value;
}
```

- To enter a new value between a range we will create a method

- `public static double readNumber(String print_text, double min, double max)`

- To calculate the mortgage

- `public static double calculateMortgage(int principal, float annualInterest, byte years){`

- To print the mortgage we will create a method

- `printMortgage(int principal, float annualInterest, byte years)`

```
public static double calculateMortgage(
    int principal,
    float annualInterest,
    byte years){

    float monthlyInterest = annualInterest / PERCENT /
MONTHS_IN_YEAR;
    int numberOfPayments = years * MONTHS_IN_YEAR;

    double mortgage = principal
        * (monthlyInterest * Math.pow(1 + monthlyInterest,
numberOfPayments))
        / (Math.pow(1 + monthlyInterest, numberOfPayments) -
1);

    return mortgage;
}
```

- To enter a new value between a range we will create a method
  - `public static double readNumber(String printText, double min, double max)`
- To calculate the mortgage
  - `public static double calculateMortgage(int principal, float annualInterest, byte years){`
- To print the mortgage we will create a method
  - `printMortgage(int principal, float annualInterest, byte years)`

```
private static void printMortgage(int principal,
float annualInterest, byte years) {
    double mortgage =
    calculateMortgage(principal, annualInterest,
years);
    String mortgageFormatted =
    NumberFormat.getCurrencyInstance().format(mor
tgage);
    System.out.println();
    System.out.println("MORTGAGE");
    System.out.println("-----");
    System.out.println("Monthly Payments: " +
mortgageFormatted);
}
```



Finally, the main changes to...

```
public static void main(String[] args) {  
    int principal = (int) readNumber("Principal: ", 1000, 1_000_000);  
    float annualInterest = (float) readNumber("Annual Interest Rate: ", 1, 30);  
    byte years = (byte) readNumber("Period (Years): ", 1, 30);  
    printMortgage(principal, annualInterest, years);  
}
```

```
import java.text.NumberFormat;
import java.util.Scanner;
public class Main {
    final static byte MONTHS_IN_YEAR = 12;
    final static byte PERCENT = 100;

    public static void main(String[] args) {...}
    private static void printMortgage(int principal, float annualInterest, byte years) {...}
    public static double readNumber(String printText, double min, double max) {...}
    public static double calculateMortgage(int principal, float annualInterest, byte years){...}
}
```