Lab#01

Objects

- Installation and setup of IDE
- What is Java, OOP, and DSA
- Variables and Datatypes
- Conditional Statements
- Iterative Statements

1. Use codespace GitHub

Step#01

Signup or login to GitHub account

Step#02

Let’s create a codespace on GitHub

https://github.com/Rafia-Shaikh-eng/Java

Note: fork this repository, so, that you can save changes on your github.

Step#03

Follow my Git account

https://github.com/Rafia-Shaikh-eng

2. Use text editor(atom) + CMD

install **JDK** and set environment variable (set path)

Install atom (text editor)

3. Use IntelliJ/Visual Studio Code IDE

_JDK, and set path

Download and install intelliJ IDE (community)

Happy coding ☺

What is Java?

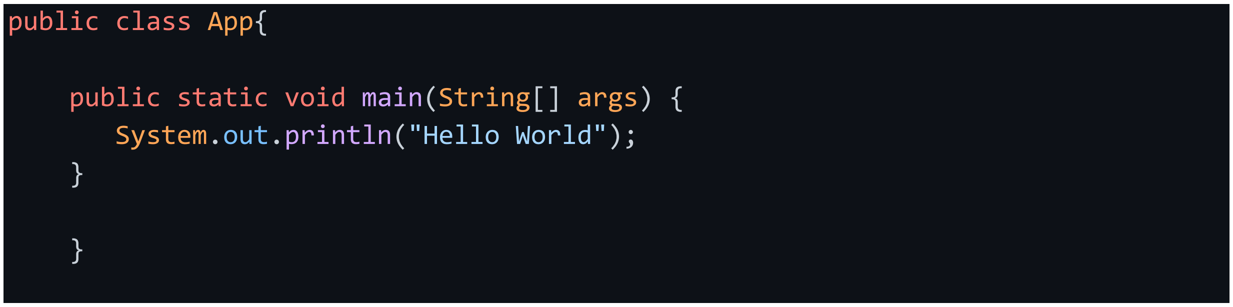
This is one of the most popular languages for beginner

Why? Java can be used to create app on the

- Mobile Applications
- Web Applications
- Desktop Applications
- Web services
- Embedded systems

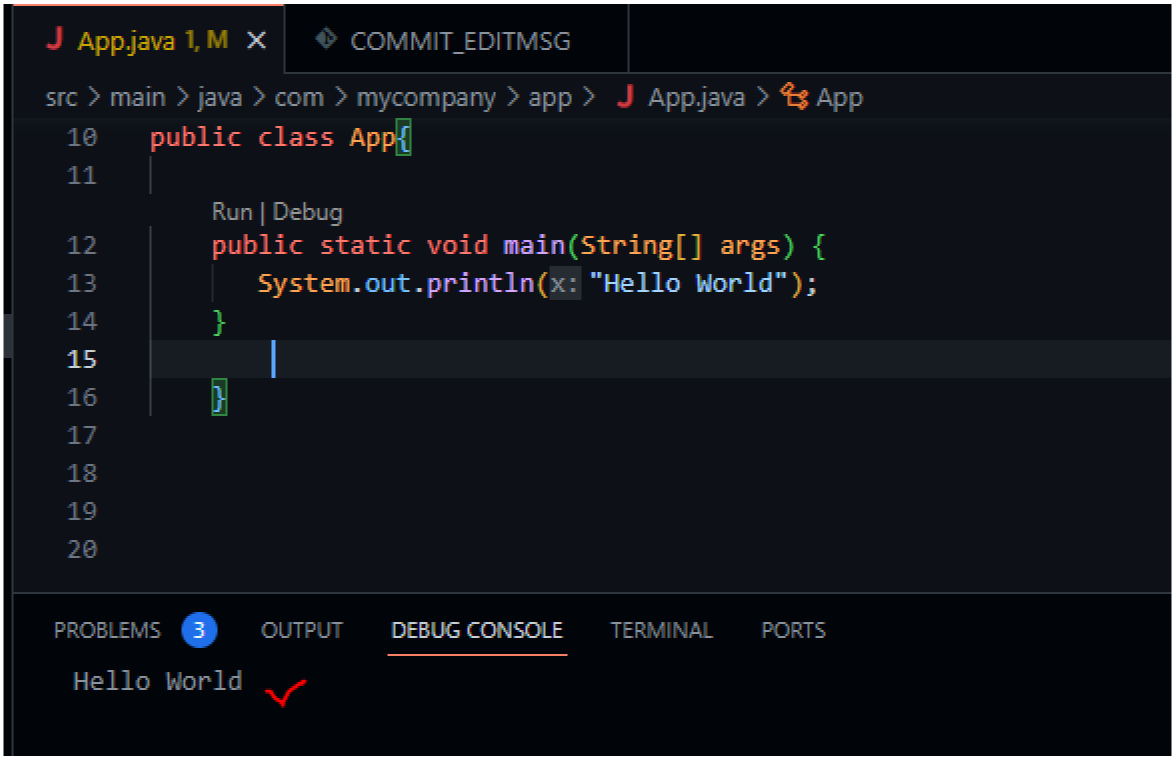
What is OOP?

Simple “Hello World” program in Java



```
public class App{

    public static void main(String[] args) {
       System.out.println("Hello World");
    }

    }
```

Output:



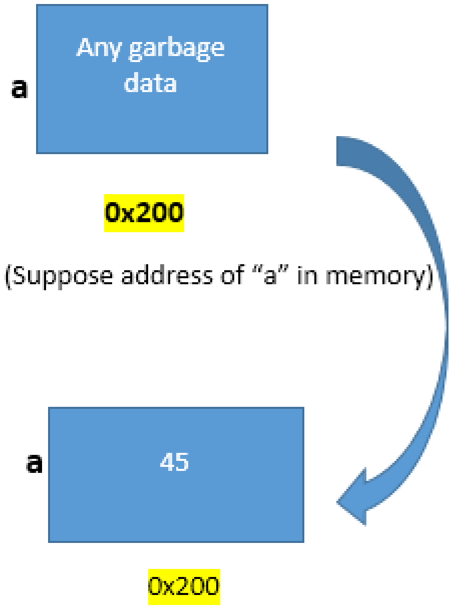
```
public class App{

    public static void main(String[] args) {
        System.out.println(x: "Hello World");
    }

    }
```

```
Hello World ✓
```

What is variable?

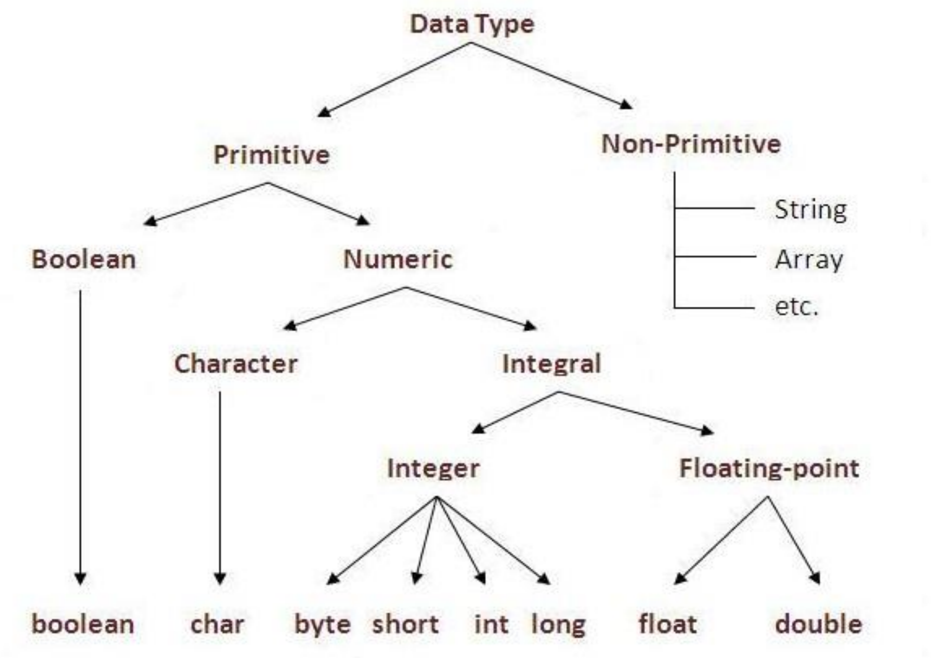
int a; (Declaration of variable);

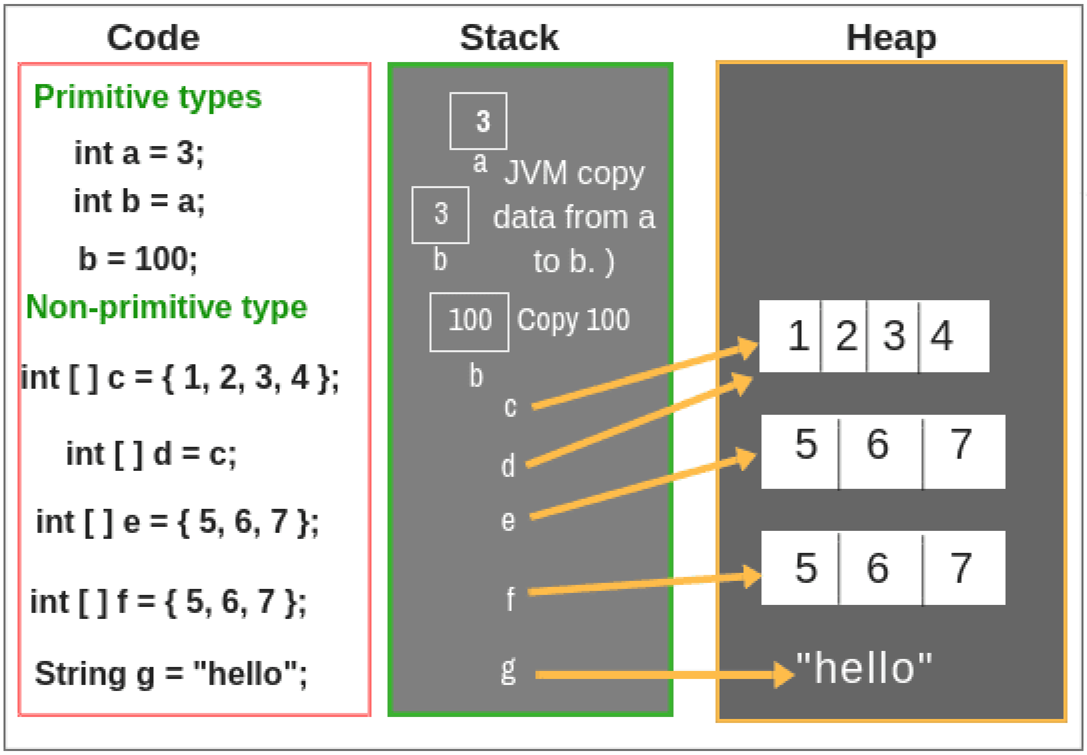
a = 45 (assign value to the variable)



Different Data types n Java

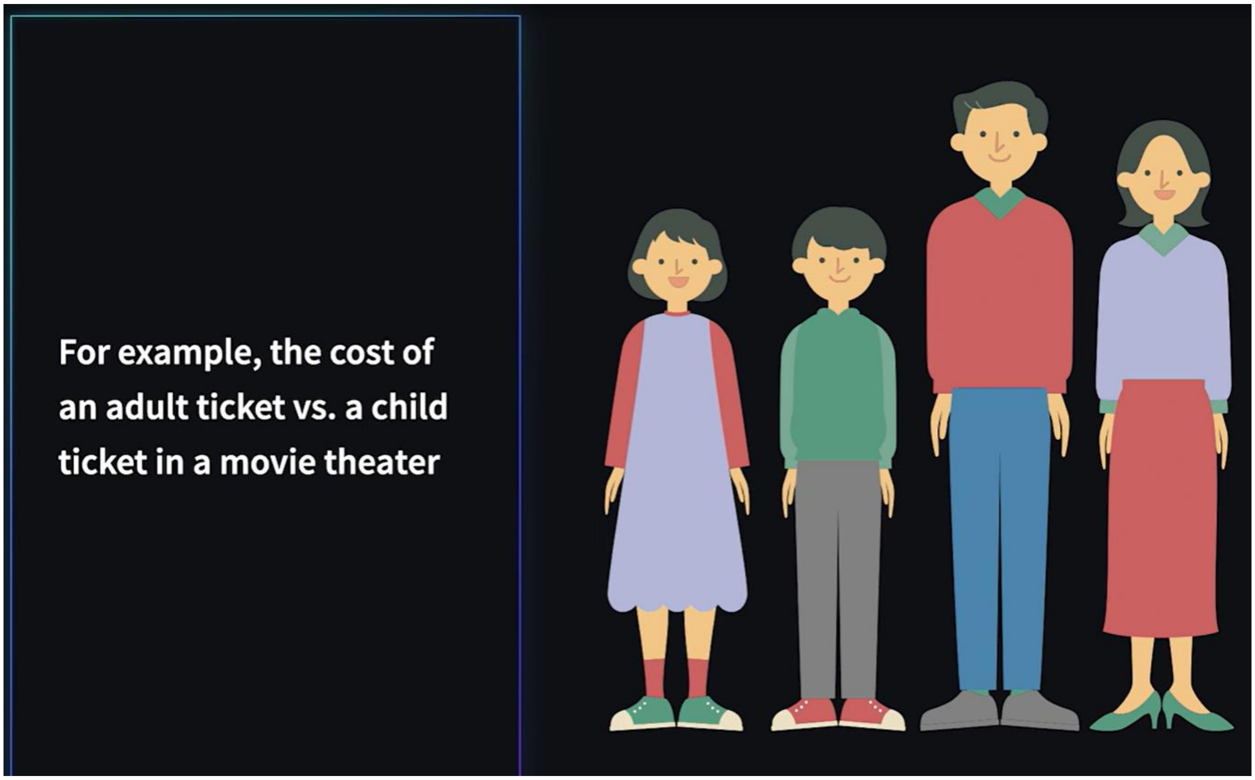
Happy coding 😊





Happy coding ☺

Control flow Statements



For example, the cost of an adult ticket vs. a child ticket in a movie theater

For adult(age>13) ticket is $10, for the child(age<13) ticket cost is $5, and there is a special discount for senior citizens(age>65) and ticket is $7 for them.

Java If-Else

The Java *if statement* is used to test the condition. It checks Boolean condition: *true* or *false*. There are various types of if statement in java.

- if statement
- if-else statement
- if-else-if
- nested if

IF Statement

If statement is the simplest decision-making statement. It is used to decide whether a certain statement or block of statements will be executed or not. The Java if statement tests the condition. It executes the *if block* if condition is true.

Syntax:

```
if(condition)
```

```
{
//code to be executed  }
```

IF-else Statement

The Java if-else statement also tests the condition. It executes the *if block* if condition is true otherwise *else block* is executed.

Syntax:

```
if(condition)
{ //code if condition is true  }
Else {
//code if condition is false  }
```

IF-else-if Statement

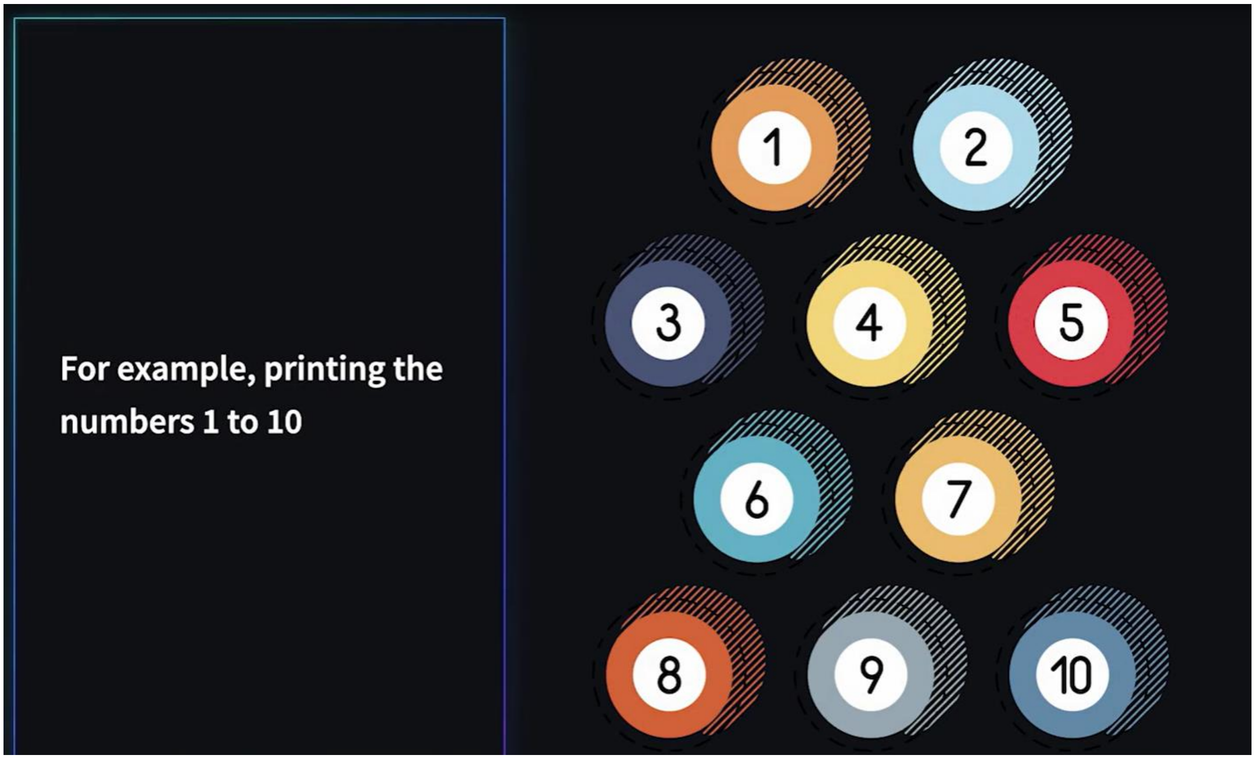
The if-else-if statement executes one condition from multiple statements.

Syntax:

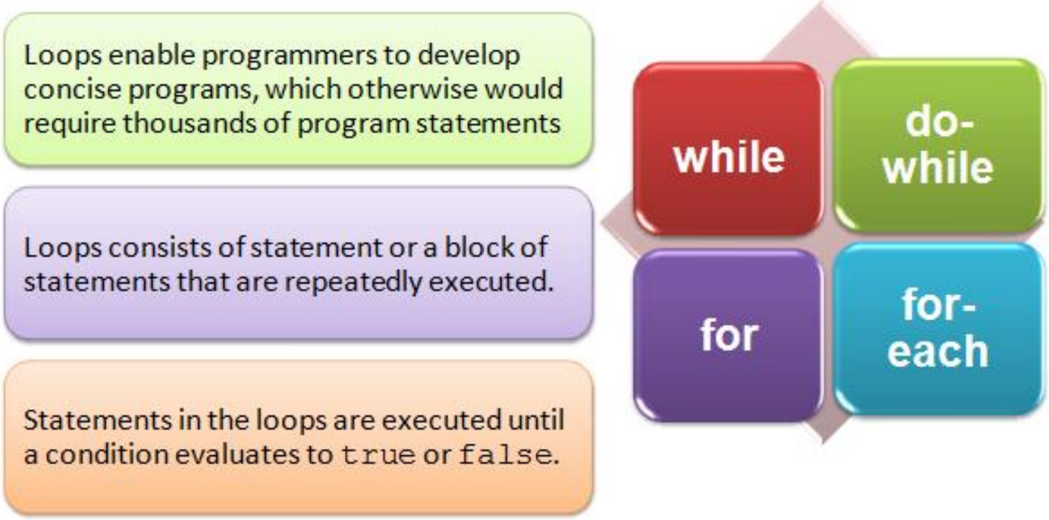
```
if(condition1) {
//code to be executed if condition1 is true  }
else if(condition2) {
//code to be executed if condition2 is true  }
else if(condition3) {
//code to be executed if condition3 is true  }
...
else {  //code to be executed if all the conditions are false  }
```

Loops:

Repetition of a part of the code is achieved by using loops.



For example, printing the numbers 1 to 10



Loops enable programmers to develop concise programs, which otherwise would require thousands of program statements

Loops consists of statement or a block of statements that are repeatedly executed.

Statements in the loops are executed until a condition evaluates to `true` or `false`.

Figure 1 Loops in Java

For loop

The initialization expression is executed only once, that is, when the loop starts.

Next, the boolean expression is evaluated and tests the loop control variable against a targeted value.

If the expression is `true`, then the body of the loop is executed and if the expression is `false`, then the loop terminates.

Lastly, the iteration portion of the loop is executed. This expression usually increments or decrements value of the control variable.

In the next iteration, again the condition section is evaluated and depending on the result of evaluation the loop is either continued or terminated.

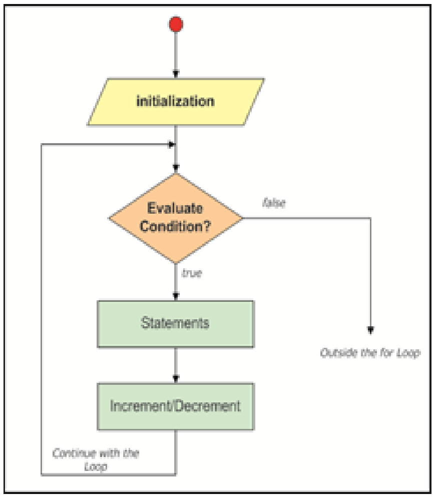


Figure 2 For Loop

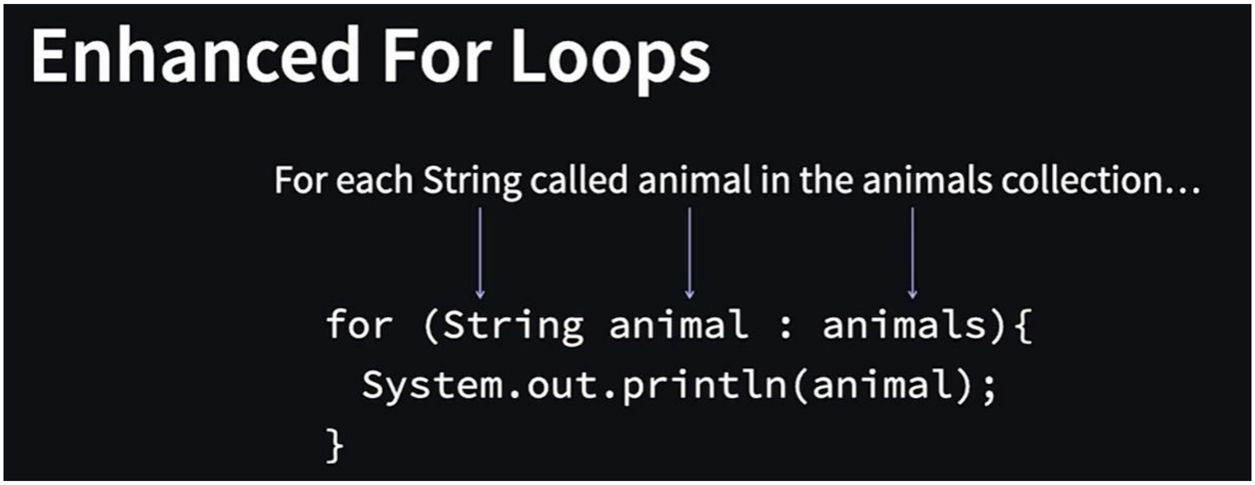
Syntax:

```
for(initialization;condition;incr/decr)
{ //code to be executed }
```

Example:

```
Class ForEx1 { public static void main(String [] args)
{
                  for(inti=1;i<=10;i++) {
System.out.println(i); }
}
}
```

Happy coding ☺

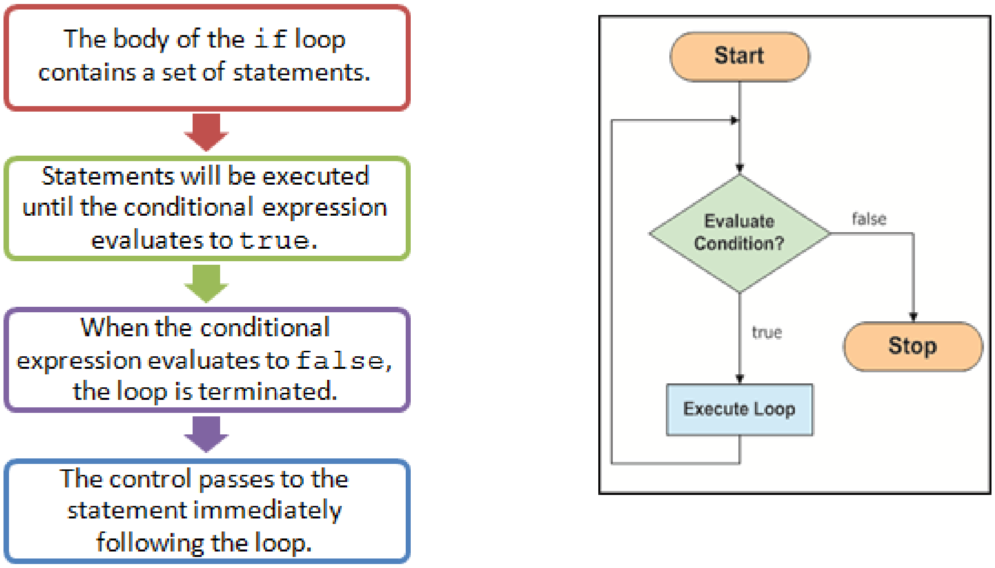


Enhanced For Loops

For each String called animal in the animals collection…

```
for (String animal : animals){
  System.out.println(animal);
}
```

While Loop:



The body of the if loop contains a set of statements.

Statements will be executed until the conditional expression evaluates to true.

When the conditional expression evaluates to false, the loop is terminated.

The control passes to the statement immediately following the loop.

Figure 3 While Loop

Syntax:

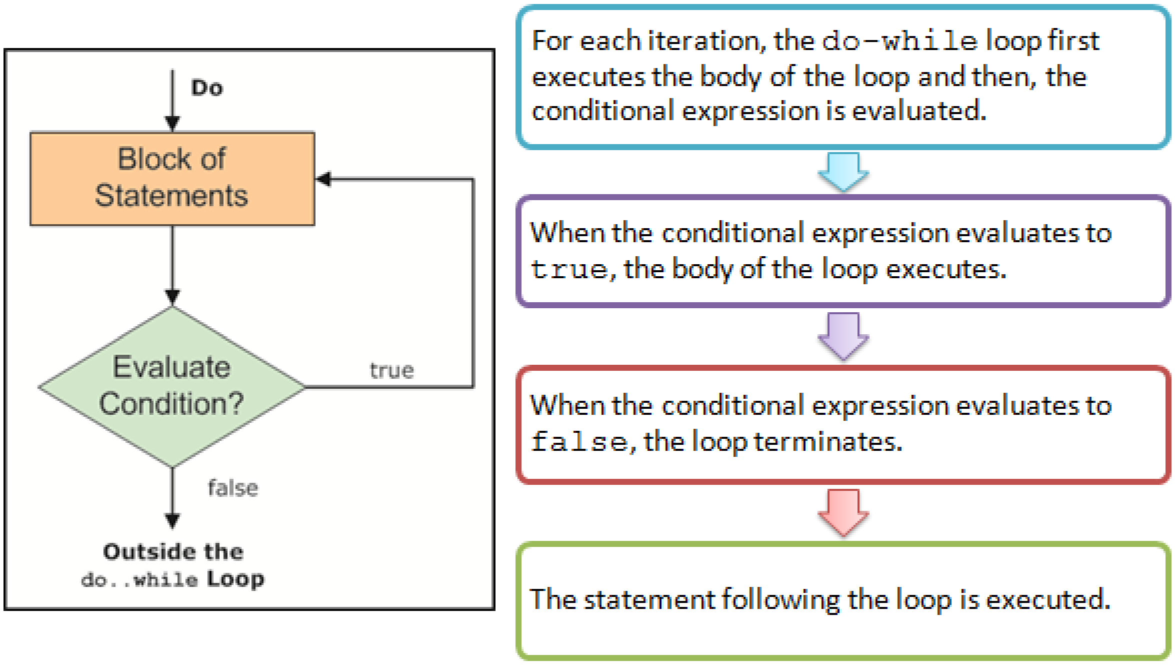
```
//initialization
while(condition)
{
//code to be executed
//increment/decrement
}
```

Happy coding 😊

Example:

```
classWhileLoopExample{
public static void main(String[] args)
{
      inti=1;
while(i<=10)
{
System.out.println(i);
i++;
}
}
}
```

Do-While loop



For each iteration, the do-while loop first executes the body of the loop and then, the conditional expression is evaluated.

When the conditional expression evaluates to true, the body of the loop executes.

When the conditional expression evaluates to false, the loop terminates.

The statement following the loop is executed.

Figure 4 Do While Loop

Syntax

```
do{
```

```
//code to be executed
}while(condition);
```

Example

```
public class DoWhileExample{
public static void main(String[] args)
{
inti=1;
do{
System.out.println(i);
i++; }
while(i<=10);}
}
```

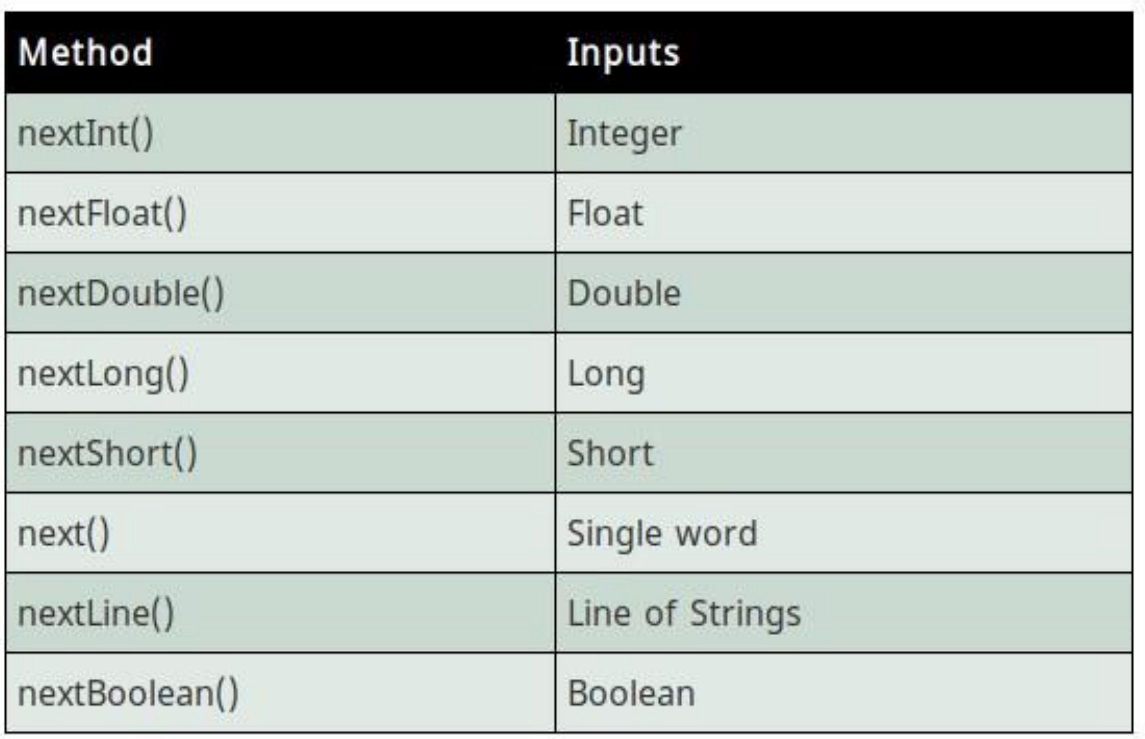
We use do-while use when the requirement is to execute the program at least once, irrespective of the condition. So even if the condition is not met, the body of the loop will execute once.

Taking Input From the User

Using Scanner Class

This is probably the most preferred method to take input. The main purpose of the Scanner class is to parse primitive types and strings using regular expressions, however it is also can be used to read input from the user in the command line.

```
import java.util.Scanner;
class GetInputFromUser
{
    Public static void main(String args[])
    {
        // Using Scanner for Getting Input from User
        Scanner in = new Scanner(System.in);
        String s = in.nextLine();
        System.out.println("You entered string "+s);
        Int a = in.nextInt();
        System.out.println("You entered integer "+a);
        float b = in.nextFloat();
        System.out.println("You entered float "+b);
    }
}
```



Method	Inputs
nextInt()	Integer
nextFloat()	Float
nextDouble()	Double
nextLong()	Long
nextShort()	Short
next()	Single word
nextLine()	Line of Strings
nextBoolean()	Boolean

Figure 5 Input Function Detail

Methods in Java:

Methods in Java are the building block of a Java application. In Java, a method is a set of code used to write the logic of the applications which perform some specific tasks or operations.

In Java programming, a method is executed when it is called from another method.

The **main()** method is the first method that is executed by **JVM (Java Virtual Machine)** in java program.

If we want to want to execute any other method, we should call it from the main() method.

When a method is called, it returns a value to the calling method. It can also perform a task without returning any value to the calling method.

A method can be called from anywhere in Java program. Each method has a single entry point and a single exit.

Happy coding 😊

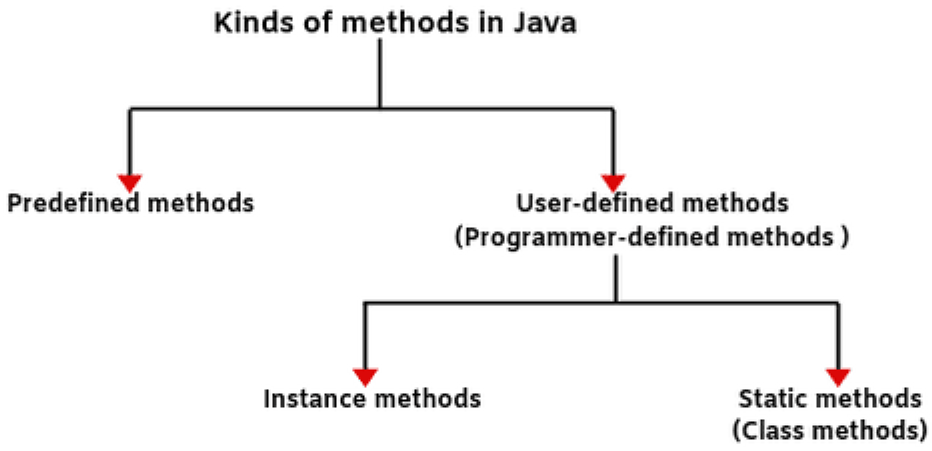


Fig: Basic kinds of methods in Java

Instance Method in Java

An instance method is used to implement behaviors of each object/instance of the class. Since the instance method represents behaviors of the objects. Therefore, instance method is linked with an object.

An instance method is also known as non-static method. So, without creating an object of the class, the methods cannot exist to perform their desired behaviors or task. It is allocated in the heap memory during the object creation.

Here is an example of an instance method declaration:

```
void m1()

{

  // This area is called an instanced area/ Non-static area.

  // logic here.

}
```

Static Method in Java

When you declare any method with a static modifier, it is called static method in java. A static method is linked with class.

Therefore, it is also known as a class method. It is used to implement the behavior of the class itself. Static methods load into the memory during class loading and before object creation.

An example of a static method declaration is as follows:

```
static void m2()

{

  // This area is called a static area.

  // logic here.

}
```

Class Activity

- Write a program which take input from the user a number and tell the user that the entered number is either positive, negative or zero..
- Write a program to print the given output using loops.

```
1

12

123

1234

12345
```

- Write a program to print the given output using loops.

```
     *

    **

   ***

  ****

 *****

******
```

Happy coding ☺

Exercise

Task#01

Implement the following classes:

1. Employee class

   Employee.java

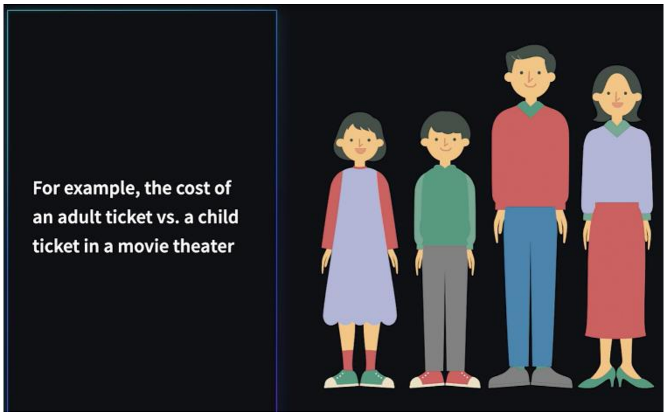
2. MenuBuilder class

   MenuBuilder.java

Task#02

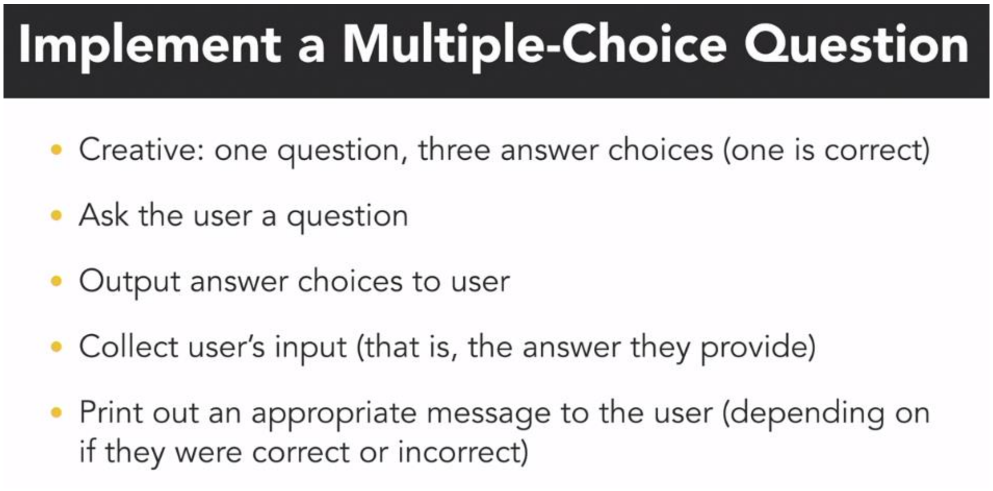
Write a java program which will take age and display the theatre ticket cost

- For adult(age>13) ticket is $10
- for the child(age<13) ticket cost is $5
- there is a special discount for senior citizens(age>65) and ticket is $7 for them



For example, the cost of an adult ticket vs. a child ticket in a movie theater

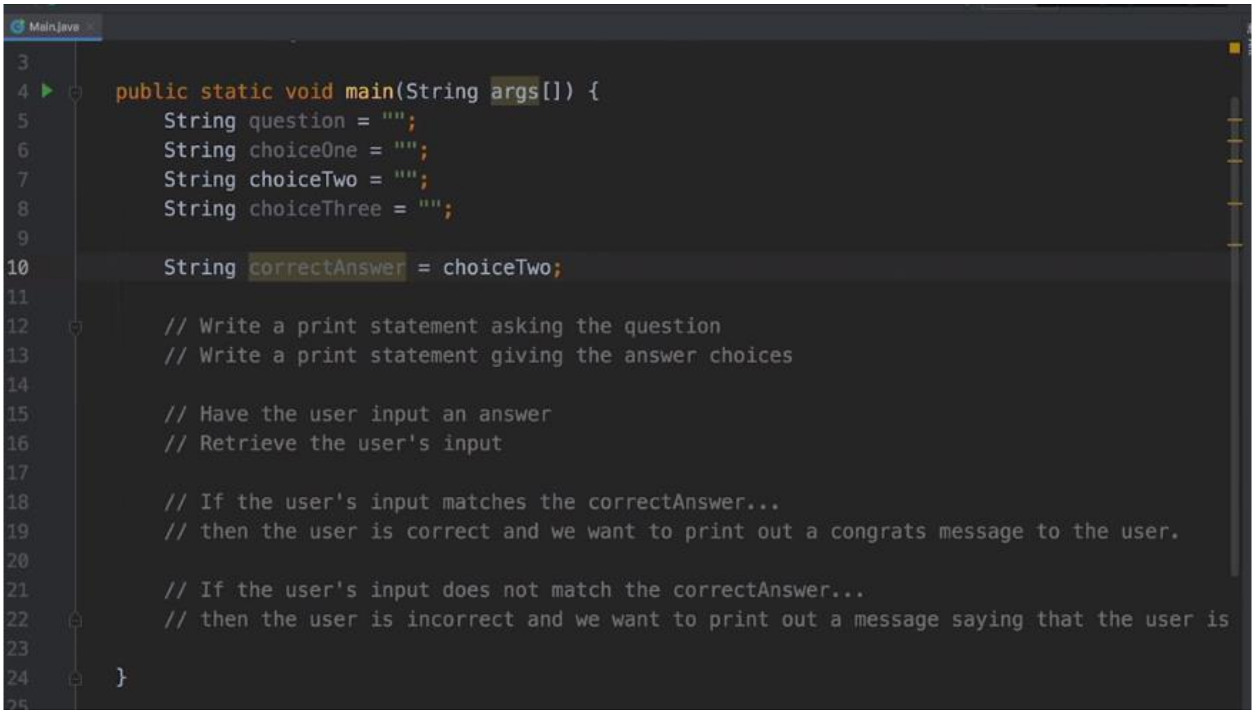
Task#03



Implement a Multiple-Choice Question

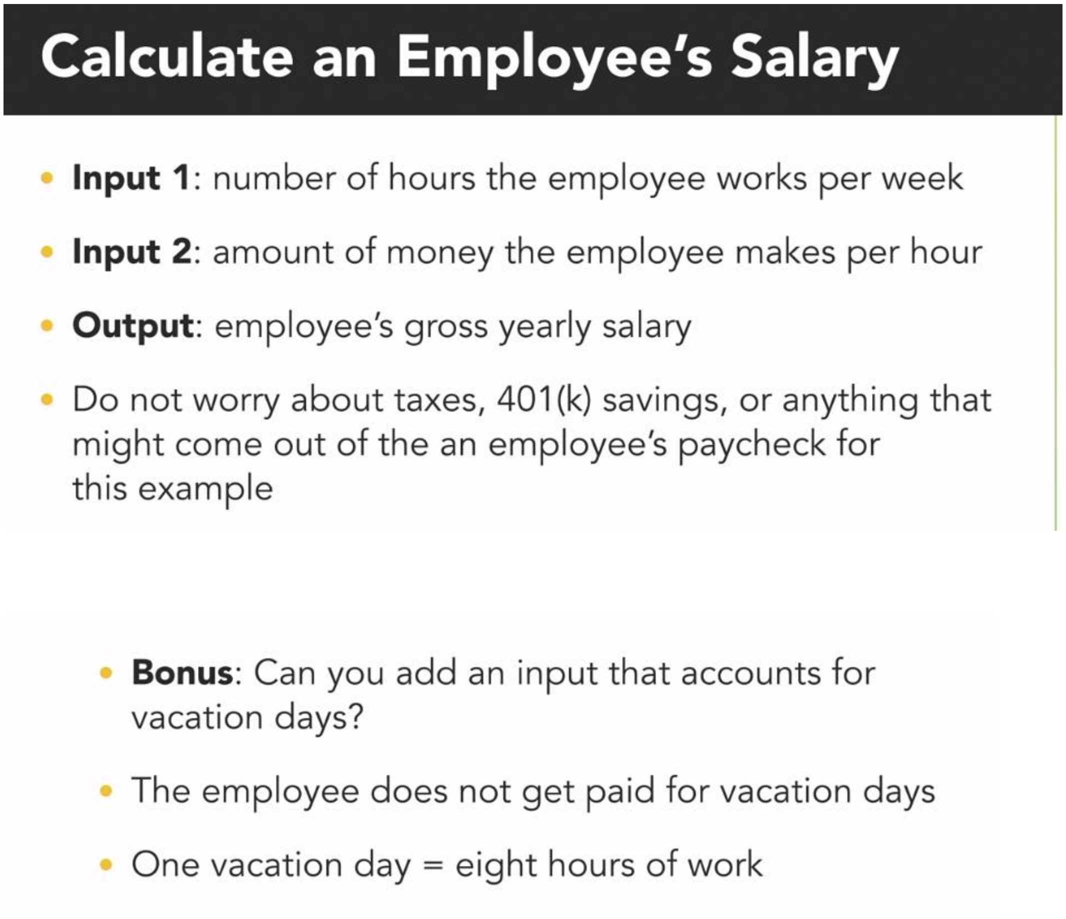
- Creative: one question, three answer choices (one is correct)
- Ask the user a question
- Output answer choices to user
- Collect user's input (that is, the answer they provide)
- Print out an appropriate message to the user (depending on if they were correct or incorrect)

Happy coding 😊



```java
public static void main(String args[]) {
    String question = "";
    String choiceOne = "";
    String choiceTwo = "";
    String choiceThree = "";

    String correctAnswer = choiceTwo;

    // Write a print statement asking the question
    // Write a print statement giving the answer choices

    // Have the user input an answer
    // Retrieve the user's input

    // If the user's input matches the correctAnswer...
    // then the user is correct and we want to print out a congrats message to the user.

    // If the user's input does not match the correctAnswer...
    // then the user is incorrect and we want to print out a message saying that the user is

}
```

Task#04



Calculate an Employee's Salary

- **Input 1**: number of hours the employee works per week
- **Input 2**: amount of money the employee makes per hour
- **Output**: employee's gross yearly salary
- Do not worry about taxes, 401(k) savings, or anything that might come out of the an employee's paycheck for this example

- **Bonus**: Can you add an input that accounts for vacation days?
- The employee does not get paid for vacation days
- One vacation day = eight hours of work

Happy coding 😊