A brief review of Java programming
Popularity of Programming Languages

Source: https://www.tiobe.com/tiobe-index/
# History of PL Popularity

<table>
<thead>
<tr>
<th>Sep 2017</th>
<th>Sep 2016</th>
<th>Change</th>
<th>Programming Language</th>
<th>Ratings</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td>Java</td>
<td>12.687%</td>
<td>-5.55%</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
<td>C</td>
<td>7.382%</td>
<td>-3.57%</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td>C++</td>
<td>5.565%</td>
<td>-1.09%</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td></td>
<td>C#</td>
<td>4.779%</td>
<td>-0.71%</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td></td>
<td>Python</td>
<td>2.983%</td>
<td>-1.32%</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>↑</td>
<td>PHP</td>
<td>2.210%</td>
<td>-0.64%</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>↓</td>
<td>JavaScript</td>
<td>2.017%</td>
<td>-0.91%</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>↑</td>
<td>Visual Basic .NET</td>
<td>1.982%</td>
<td>-0.36%</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>↑</td>
<td>Perl</td>
<td>1.952%</td>
<td>-0.38%</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>↑</td>
<td>Ruby</td>
<td>1.933%</td>
<td>-0.03%</td>
</tr>
<tr>
<td>11</td>
<td>18</td>
<td>↑</td>
<td>R</td>
<td>1.816%</td>
<td>+0.13%</td>
</tr>
</tbody>
</table>
Java is also needed in jobs.
About Java

Java is

- One of the most popular languages in the past years: Simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, multi-threaded, dynamic, and more.

Three main elements: Class, Type, and Object

- An object is the basic unit in Java
- A class defines the type of an object
Java Programming Basics

- A class consists of
  - fields (to store data)
  - methods (to define operations that can act on data)

```java
public class Hello {
    public static int var;
    public static void say(String s) {
        System.out.print("Hello "+s);
    }
    public static void main(String[] argv) {
        say("World!");
    }
}
```

The class name (Save this code as Hello.java)

- `public class Hello` - The class name (Save this code as Hello.java)
- `public static int var;` - A field
- `public static void say(String s) {` - A method
- `public static void main(String[] argv) {` - The main method (The entry point while executing the program)
Modifiers

“public” indicates that anyone can run/extend/import this class

```java
public class Hello {
    public static int var;
    public static void say(String s) {
        System.out.print("Hello "+s);
    }
    public static void main(String[] argv) {
        say("World!");
    }
}
```

“static” indicates the field/method belongs to the class, not objects

“void” indicates that the method returns nothing
How Java works

- Execute your code in command lines
  - “javac Hello.java” to generate Hello.class
  - “java Hello” to execute the bytecode
Example: Operator

- Operators are similar to C++
  - E.g., =, +, -, *, /, %

- A simple example:

- Sum 1 to 100 using a formula

```java
public class Example {
    public static void main(String[] argv) {
        int n = 100;
        System.out.println("1+2+...+" + n + " = " + (n * (n + 1) / 2));
    }
}
```

```
javac Example.java
java Example

1+2+...+100 = 5050
```
Example: Loop

- Sum 1 to 100 using a method with for-loop

```java
public class Example {
    public static int sum(int n) {
        return n*(n+1)/2;
    }
    public static void main(String[] argv) {
        int n = 100;
        System.out.println("1+2+...+"+n+" = " + sum(n));
    }
}
```

```bash
javadoc Example.java
java Example
1+2+...+100 = 5050
```
Example: Loop

- Sum 1 to 100 using a method with for-loop

```java
public class Example {
    public static int sum(int n) {
        int total = 0;
        for (int i = 1; i <= n; i++) {
            total += i;
        }
        return total;
    }
    public static void main(String[] argv) {
        int n1 = 100, n2=200;
        System.out.println("1+2+...+" + n + " = " + sum(n));
    }
}
```

```
javac Example.java
java Example
1+2+...+100 = 5050
```
Example: Loop

- Sum 1 to 100 using a method with for-loop

```java
public class Example {
    public static int sum(int n) {
        int total = 0;
        for (int i = 1; i <= n; i++) { total += i; }
        return total;
    }

    public static int sum2(int n1, int n2) {
        int total = 0;
        for (int i = n1; i <= n2; i++) { total += i; }
        return total;
    }

    public static void main(String[] argv) {
        int n1 = 100, n2 = 200;
        System.out.println(n1+"+...+"+n2+" = " + sum2(n1,n2));
    }
}
```

javac Example.java
java Example

1+2+...+100 = 5050
Use pre-defined class library

- Use java.util.Scanner for getting inputs

- The Scanner class reads the input stream and divides it into tokens by delimiters (whitespace)

- The Scanner class includes the following methods:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hasNext()</td>
<td>Return true if there is another token</td>
</tr>
<tr>
<td>next()</td>
<td>Return the next token</td>
</tr>
<tr>
<td>hasNextType()</td>
<td>Return true if there is another token that can be interpreted as the <strong>Type</strong></td>
</tr>
<tr>
<td>nextType()</td>
<td>Return the next token that can be interpreted as the <strong>Type</strong></td>
</tr>
</tbody>
</table>
Use pre-defined class library

- Import the package
  
  ```java
  import java.util.Scanner;
  ```

- Construct a Scanner object:
  
  ```java
  Scanner in = new Scanner(System.in);
  ```

- Call its method:
  
  e.g., `in.nextInt()` or `in.hasNext()`
Example: Get a user input

- Sum using java.util.Scanner class

```java
import java.util.Scanner;
public class Example {
    public static int sum(int n) {
        int total = 0;
        for (int i = 1; i <= n; i++) {
            total += i;
        }
        return total;
    }
    public static void main(String[] argv) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter n: ");
        int n = in.nextInt();
        System.out.println("1+2+...+" + n + " = " + sum(n));
    }
}
```

```
javac Example.java
java Example
Enter n: 100
1+2+...+100 = 5050
```
Example: Get user inputs

- Sum using java.util.Scanner class

```java
import java.util.Scanner;
public class Example {
    public static int sum(int n) {
        int total = 0;
        for (int i = 1; i <= n; i++) { total += i; }
        return total;
    }
    public static int sum(int n1, n2) {
        int total = 0;
        for (int i = n1; i <= n2; i++) { total += i; }
        return total;
    }
    public static void main(String[] argv) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter n1 n2: ");
        int n1 = in.nextInt();
        int n2 = in.nextInt();
        System.out.println(n1+"...+"+n2+" = " + sum(n1,n2));
    }
}
```

javac Example.java
java Example

Enter n1 n2: 10 100
10+...+100 = 5005
About Eclipse

Eclipse is

- An Integrated Development Environment (IDE) for Java and also many other languages
- An open source platform (free!)
- Maintained by many software development leaders like IBM and Borland
Eclipse Extension

Furthermore, Eclipse

- provides a common environment that companies can modify and customize by creating plug-ins

- These plug-ins can add functionality to Eclipse like modeling, UML, XML, metrics, reliability reports, and other information.

- The Eclipse web site has a list of links to many popular plug-in repositories
Learn Eclipse and Java

- Eclipse and Java tutorials. Watch this if you are a total beginner.
  http://ecliptutorial.sourceforge.net/index.html

- A nice introduction to eclipse by L. Williams et al. NCSU.
  http://realsearchgroup.org/SEMaterials/tutorials/eclipse/eclipse_tutorial_3.5.html

- A nice java/eclipse tutorial on youtube:
  https://www.youtube.com/watch?v=r59xYe3Vyks&list=PLS1Qu1Wo1RIbfTjQvTdj8Y6yyq4R7g-A1
Homework 1 (Due on 9/27)

- BMI Calculator:
  - BMI = ( Weight in Kilograms / ( Height in Meters x Height in Meters ) )

- Enter Height and Weight, return BMI and
  - “You are not in shape. Actually, you are not even close.” if BMI >= 30
  - “To be honest, you are not in shape.” if 30 > BMI >= 26
  - “You are in shape” if 26 > BMI >= 20
  - “You are under shape” if 20 > BMI

- Use Eclipse to write/execute/debug your java code

- Upload your code using WM5 (no direct copy accepted)

- TAs will show you “clear” hints to do so in Tuesday’s lab (4:00-6:00pm, Sep. 25)
Coming up…

- Happy Moon Festival!

- The first lab is scheduled on Tuesday, Sep 25, 4:00-6:00pm. TAs will talk about Eclipse and HW1 (formats for automatic grading)

- Other labs are scheduled on Monday 12:00-2:00pm.

- We will discuss object oriented design and abstract data type next week

- Read TB Chapter 1 and Chapter 2