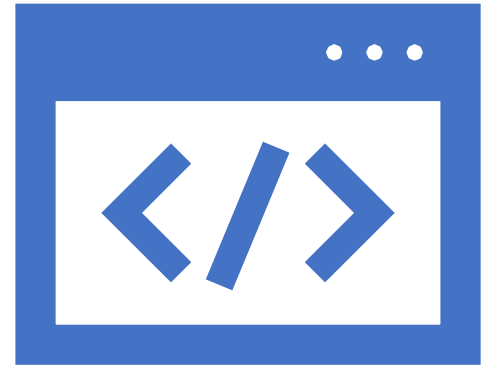


Programming with Java

Noob programming and starting problem solving



F S T M K U I S

ABOUT ME

Name: Arif Zuhairi bin Mohd Basri

Age: 24

- Semester 6 BCNT(formerly BSE)
- Exco Jabatan SK PMFSTM 19/20
- From Windows ME, 95, XP, to Linux, playing FIFA 93,95-07,10, to Programming
- Currently exploring Machine Learning and Cloud Computing(AWS, Google Cloud)

Exp:

- General IT: 7 years(study, work, hobby)
- Java: 1 year+
- Python(main lang)
- Php, C++(years ago)

Agenda

- **Hardware & Software:** Computer & Integrated Development Editor(IDE)
- **Problem & Solution:** IDEAL model
- **Java?:** Purpose, history
- **Hello World:** Basic syntax, JVM, install & run
- **Code Structure:**
- **Data:** Data types, literals & conversion
- **Operation:**
 - Basic arithmetic
 - Branching
 - Repetition
 - IO

Problem Solving Skills

- **Analytical thinking**

- Evaluate and make decisions
- Use logical and methodical approach

- **Lateral thinking**

- Creative and out-of-box thinking
- Discard obvious, skip traditional thinking, ignore preconceptions

- **Team**

- Key component in problem solving
- Not necessary analytical/lateral skills: e.g.: management, communication and negotiation

IDEAL SOLUTION MODEL

Identify

Define

Explore

Act

Look

Identify Issues



Investigate causes until the root cause



Gather relevant information



Break problems into parts

Define Goals

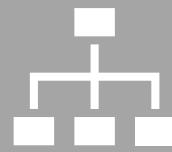
- Set target output or solution
- Ideation process
- SMART model

Specific	Measurable	Achievable	Relevant	Time bound
<ul style="list-style-type: none">• Goal must be clear	<ul style="list-style-type: none">• Ability to track, stay focused	<ul style="list-style-type: none">• Realistic and attainable	<ul style="list-style-type: none">• Care on progress	<ul style="list-style-type: none">• Target date and priority

Explore options



Explore and prepare solutions draft



Use presentation medium(flow chart, pseudo-code, story board)



Decide on final and best solution

Act on best
solution



To-do list



Build-test-Repeat



Progress monitoring

Type of Solutions

Algorithmic Solution

- Can be solved by completing a series of actions in steps

Heuristic Solution

- Cannot be reached by a direct set of steps (require reasoning built on knowledge and experience, and a process of trial and error)

Algorithm

- A specific and step-by-step set of instructions for carrying out a procedure or solving a problem, usually with the requirement that the procedure terminate at some point
- 3 types of algorithm representation will be explained:
 - IPO chart
 - Flowchart
 - Pseudocode

Algorithm Development Guidelines

Identify

Identify the input and output of the problem.



Use

If necessary, use 'Divide & Conquer' strategy to decompose the problem into smaller and manageable sub problems. Decompose the problem until all the sub problems can be solved to get the required results



Identify
and list
out

For each sub problem, identify and list out the steps involved in solving it

Look and Learn



Working smoothly?



Improvement?

Difficulties with Problem Solving



Claim have not been taught to solve problem



Afraid to make the wrong decision



Not going through the problem-solving steps thoroughly



The most difficult task is writing the instruction (How to determine the largest number from a group of three numbers)



JAVA: A Cup of Coffee

Java is a general-purpose, concurrent, object-oriented, class-based, and the runtime environment(JRE) which consists of JVM which is the cornerstone of the Java platform.

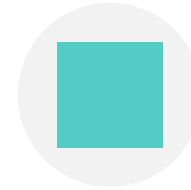
What is JAVA used for?



BANKING: TO DEAL WITH
TRANSACTION
MANAGEMENT.



RETAIL: BILLING
APPLICATIONS THAT
YOU SEE IN A
STORE/RESTAURANT



IT: JAVA IS DESIGNED TO
SOLVE
IMPLEMENTATION
DEPENDENCIES.



ANDROID:
APPLICATIONS ARE
EITHER WRITTEN IN JAVA
OR USE JAVA API.



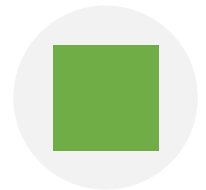
FINANCIAL SERVICES:
SERVER-SIDE
APPLICATIONS.



STOCK MARKET: TO
WRITE ALGORITHMS AS
TO WHICH COMPANY
THEY SHOULD INVEST IN.



BIG DATA: HADOOP
MAPREDUCE
FRAMEWORK



SCIENTIFIC AND
RESEARCH COMMUNITY:
TO DEAL WITH HUGE
AMOUNT OF DATA.

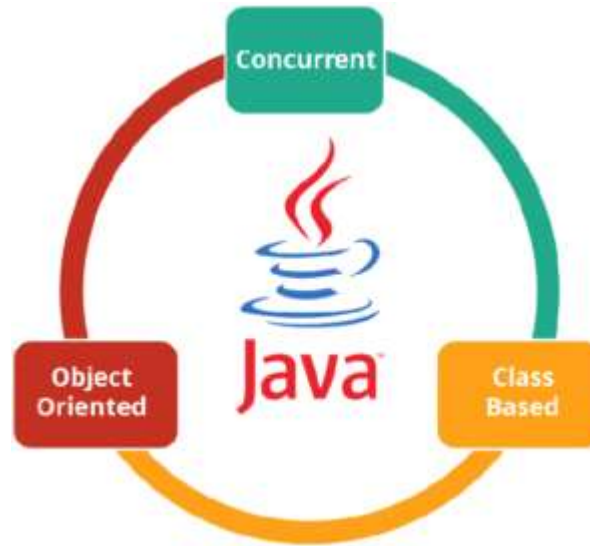
History

- Initially named Oak Sun Microsystem set top box (1991)
- Rename to Java after Oak Technology (1994)
- Versions :
- Java Standard Edition (Java/J2SE/JavaSE)
- Java Enterprise Edition (J2EE/JavaEE JakartaEE)
- Java Micro Edition (J2ME) (CDLC)
- Java Card

Green Team in 1995 for Sun Microsystems

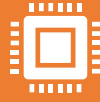


What is Java?



Credit to: Edureka

Java Features



Simple: Java has made life easier by removing all the complexities such as pointers, operator overloading as you see in C++



Portable: Java is platform independent



Object-oriented: Distributed, encapsulated, message passing.

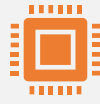


Secured: All the code is converted in bytecode after compilation, which is not readable by a human. and java does not use an explicit pointer and run the programs inside the sandbox



Dynamic: It has the ability to adapt to an evolving environment which supports dynamic memory allocation

Java Features



Distributed: Java provides a feature which helps to create distributed applications. Using Remote Method Invocation (RMI)



Robust: Java has a strong memory management system.



High Performance: With the use of JIT (Just-In-Time) compilers, Java enables high performance.



Interpreted: Java is compiled to bytecodes, which are interpreted by a Java run-time environment.



Multithreaded: Java supports multiple threads of execution (a.k.a., lightweight processes)

Java Virtual Machines(JVM)

- Run Java program
- Run non Java language then compile to Java bytecode

Class loader

- Load all classes which being use by a Java program
- Verify import
- Allocate memory
- Initialize classes and variables and invoke main class

Just In Time compiler

: Translate Java bytecode into machine language to speed up execution

Heap

: Memory area that allocated for direct memory location

Java Runtime Environment(JRE)

- JRE refers to a runtime environment in which Java bytecode can be executed. It implements the JVM (Java Virtual Machine) and provides all the class libraries and other support files that JVM uses at runtime. So JRE is a software package that contains what is required to run a Java program. Basically, it's an implementation of the JVM which physically exists.

Java Development Kit(JDK)

It is the tool necessary to:-

- Compile
- Document
- Package Java programs.

Other JVM Languages

Python

JS

Ruby

Kotlin

Perl

R

Installation

- Hands-On

General Syntax

- Hands-On

Hello World

- Hands-On

Construction

- Hands-On

Variable

- Hands-On

Basic Data Type

- Hands-On

Literals

- Hands-On

Wrapper

- Hands-On

Data Conversion

- Hands-On

Data Operators

- Hands-On

Common Data Structures - Array

- Hands-On

Method

- Hands-On

Control statements - Selection

- Hands-On

Control statements - Iteration

- Hands-On

Nested

- Hands-On

Basic Encapsulation

- Hands-On

Beginner Resources

- <https://www.edureka.co/blog/java-tutorial/>
- <https://www.javatpoint.com/java-tutorial>
- <https://www.geeksforgeeks.org/java/>
- <https://docs.oracle.com/javase/8/docs/>

Other interesting resources:

- <https://introcs.cs.princeton.edu/java/home>
- <http://www.java2novice.com/>
- <http://www.learntosolveit.com/java/>



A little challenge:

- <https://codingbat.com/java>
- <https://www.hackerrank.com/>
- <https://leetcode.com/>

Don't forget:

- DuckDuckGo / Google
- Stackoverflow

Code repositories:

- <http://github.com/>

freeCodeCamp (🔥)

edureka!

”

THANK YOU

”

twitter.com/AreRex14

Telegram: [@AreRex14](#)

sig-fstm.github.io