**JAVA BASICS**

.java(source) -> jdk(jre/jvm) – compilation

.class(bytecode) – compiler generated file – not human understandable

JVM (Java Virtual Machine)

* No physical existence
* Abstract in nature
* Loads the code, Verifies the code, executes the code -> Runtime Env
* Platform dependent

JRE (Java Runtime Environment)

* Physically exists
* It is an implementation of JVM
* Platform dependent
* Minimum requirement for executing any java code

JDK (Java Development Kit)

* It is a full featured standard development kit
* JRE + Developer Tools
* To develop and execute java code, minimum requirement is JDK

Compilation – is always with file extension (Ex: javac HelloWorld.java)

Execution – is always without file extension (Ex: java HelloWorld)

Class name ===== File name

CLASS -

* Blueprint from which individual objects are created
* Defines a new datatype
* Combination of properties(members) and behavior(methods)

public class SmartPhone {

storage,screensize,model,manufacturer -> properties/members

talk(),clickpic(),recording() -> functions/methods

}

OBJECTS –

* Instances of class

Iphone – storage,screensize,model, manufacturer

Samsung s3 - storage,screensize,model, manufacturer

Nokia - storage,screensize,model, manufacturer

Redmi - storage,screensize,model, manufacturer

Class specific - Members/Properties

5 pdf files – 2 Technical / 3 nontechnical -> 1 English 1 Tamil 1 Kannada

3 docx

3 mp3

6 txt

5 html

10 java

MAINTAINENCE

100 java files – use sub-folders to place these files

FOLDER / PACKAGE

Eclipse – Tool/IDE(Integrated Development Environment)

Workspace – physical folder location

Perspective – collection of different views(default – Java)

Views – different portions/tabs of the perspective

Create Java Project – name – jdk1.8 - finish

Right click on src folder - > Specify the package first and then the class name -> next -> finish

Java Project gets created on the left-side(Package Explorer)

Class/Interface

API?

Interface – User n Program, enables interaction between different components in the application

Java API – go through the docs

<https://docs.oracle.com/javase/7/docs/api/>

HISTORY

James Gosling – early 1990’s – Oak (Tree)

Java1.0 May 20,1995, Sun Microsystems

.

.

.jdk1.5 – Java 5.0

.

Jdk1.8 – Java8.0 (stable)

Java – statically typed language – int x=10;

char c=’h’;

String str=”string”;

Datatypes – Primitive(8) and Non-Primitive

Primitive(8) –numeric - int, double, float,long, byte, short

non-numeric- char,boolean

Non-Primitive – String, Arrays, User-defined classes

Primitive – Stack memory

Non-Primitive – Heap memory

Identifier/Variable

int number1=10;

camel-case

Type Conversions –

1. Automatic/Widening

Byte -> Short -> Int -> Long -> Float -> Double

1. Type-casting/Narrowing

Double -> Float -> Long -> Int -> Short -> Byte

CLA (Command Line Arguments)

main(**String args[**])

args[0]=1

args[1]=10

args[2]=100

Wrapper Classes –

int – Integer Integer.parseInt(String) – string to int

Integer.valueOf(String) – string to int

double – Double

char – Char

Autoboxing - primitive to non-primitive

and Unboxing – non-primitive to primitive

Assignment –

1. Think of different classes and objects in real-time and list it down
2. Attach source files to Eclipse
3. Understand stack and heap memory wrt datatypes
4. Go through Java API documentation
5. Java Coding Conventions – do a complete reading and follow it
6. Develop a program for playing guessing game.
7. Demonstrate Autoboxing and Unboxing through an example program.