Day 1 – Orientation & Setting up the Environment (Downloaded many software and installed it)

Day 2 – Unix OS, Unix Commands [ ls, mkdir, cp, cd, rm, rm -r, rm -f, touch, cat, sudo, chmod, echo, vi, date, time, cal, pwd, user, whoami ….]

Day 3 – File handlings, Shell Scripting, creating files, updating files, deleting files. Waterfall Vs Agile

Day 4 – SDLC, Java Fundamentals, HelloWorld Program, Git

Day 5 :

Revisit Java concepts

Deep Dive into Git & Github [ Creating repo, understanding types of repo, committing, push, pull, resolving merge conflict …]

Wrapper Class, OOP Concepts, String, System package, Packages, Constructors, Control Flow statements

Revisit of JAVA

* JAVA is a High Level, Platform independent, multi-threaded, object-oriented prog. Lang.
* Java Uses Class & Object concepts
* Java eliminates some complex concepts from C lang (like manual memory handling, pointer manipulation, multiple inheritance etc.,)
* Java supports many OOP features like Abstraction, Polymorphism, Inheritance & Encapsulation
* Java is complied & interpreted prog. Lang.
* Editions – Standard Edition (SE), Enterprise Edition(EE), Micro/Mobile Edition (ME)
* Java Environments – JDK (Java Development Kit), JRE (Java Runtime Environment)
* Class – Is a blueprint for objects, Object – Instance of a class
* Java is a Case sensitive lang & Strongly typed lang.

Learning Steps

1. All supported characters, symbols & operators
2. All keywords (reserved words)
3. How to write Expressions, functions
4. How to write class, how to create objects, how to write methods
5. OOP implementation

A-Z, a-z,0-9 +,-,\*,/,%, ++, --, ., ;, {}, (), [], ::, &, |, !, ->, =, ==, +=, -=, \*=, /=, ^, &&, <<, >>, ? :; !=,

Java supports 8 primitive data type (Raw data types)

|  |  |  |  |
| --- | --- | --- | --- |
| Sl No | Data Type | Range or Size | Wrapper Class |
| 1 | boolean | true/false (1 bit – 0/1) | Boolean |
| 2 | byte | 8 bits or 1 Byte | Byte |
| 3 | char | 16 bits or 2 bytes | Character |
| 4 | short | 16 bits or 2 bytes | Short |
| 5 | int | 32 bits or 4 bytes | Integer |
| 6 | float | 32 bits or 4 bytes | Float |
| 7 | long | 64 bits or 8 bytes | Long |
| 8 | double | 64 bits or 8 bytes | Double |

Primitive data types are non-objects . Bcos of this reason Java is not a pure OOP Lang.

IDE – Integrated Development Environment

Eclipse EE IDE. (Open Source IDE) – Popular IDE for Java Development

When opening Eclipse for the First Time, it will ask for a workspace.

Workspace is a folder in hard disk where all the projects created in the eclipse will be saved.

Use Ctrl+Space – Shortcut for auto suggestion

Sysout + Ctrl + Space == Automatically updated to System.***out***.println();

Ctrl+F11 = To Run current Java Program

For Java Project which is created in eclipse, src – is the input folder and bin – is the output folder.

com.revature = com is the main folder and revature is the sub-folder

comments in JAVA

1. Single Line Comment // (It comments a single line only)
2. Multi Line Comment /\* \*/ (It comments multiple lines continuously
3. Documentation Comment /\*\* \*/ (It is used to generate html document [API documentation] from the code using Javadoc tool)

<https://github.com/syskantechnosoft/Batch4JavawithAngular>

Functions Syntax

Access\_Specifier/modifier return\_type function\_name (list of arguments with it’s data type) {

Function body;

}

Example

public void display(String result) {

System.out.println(“Result =” + result);

}

Functions written inside a class is called method.

Class = Properties & Methods [ state & behavior ]

Types of methods

1. No Argument No Return (It will not return anything, return type is void) – public void show() { }
2. No Argument with return (Arguments will not be passed but has a return type) – public int show() { }
3. With Argument No return --- public void show (int result) { }
4. With Argument with return -- public float show (int years, float interest) { }

A method can return any type of data (primitive data type or derived data type)

Java consist of 8 primitive data types (boolean, byte, short, char, int, float, double, long)

Primitive data type is also called as built-in data type/ system data types

Derived Data type / Custom data types / User-defined data types [ arrays, list, class, enums, interface ]

**package** com.revature;

**public** **class** MethodDemo {

**public** **void** display() {

System.***out***.println("This is No Argument No return Method!!!");

}

**public** String display1() {

**return** "This is No Argument with return method!!!";

}

**public** **void** display(**int** result) {

System.***out***.println("This is with argument and No return method!!! Result is ="+result);

}

**public** String display(**int** a, **int** b) {

**return** "This is with argument and with return method!!!! result ="+(a+b);

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

MethodDemo obj1 = **new** MethodDemo();

obj1.display();

System.***out***.println(obj1.display1());

obj1.display(25);

System.***out***.println(obj1.display(5, 10));

}

}

Constructor – It’s a special method which will have same name as the class name

Employee -- Employee()

Types of Class

1. Simple Class – A class with properties and methods
2. POJO Class – (Plain Old Java Object) A class which is not extending other class nor implementing any interface
3. Concrete Class – A class with no abstract methods
4. Abstract Class / In-complete/Non-concrete class = A class with one or more abstract methods
5. Bean Class = A class with properties & getter,setters method
6. Entity Bean class = A bean class which represents database table
7. Wrapper class = A class used to convert primitive data type to it’s corresponding object notation
8. System Class/ Built-in Class = Class which is defined by the developer [ String, StringBuffer, StringBuilder etc.,]
9. User defined Class /Custom class = A class defined by the developer or end-user

Types of variables

1. Primitive variables (int, float, double, char, boolean, short, byte, long)
2. Derived variables / custom variables / user defined variables ( arrays, collections, enums, objects)

Example :

int a = 25; // variable declaration & initialization

int b; // variable declaration only

b=30; // assigning value to variable

array

int a[] = new int[10]; // derived data type [homogenous array & heterogeneous array]

Method inside another method == Recursive method (Recursion)

JAVA Lang Fundamentals

1. Data type
2. Variable declaration
3. Operators [+,-, ++, --, ==, !=, <, <=, >, >=, &&, ||, !, <<, >>, ?:;]
4. Statements

Y=mx+c // problem statement

In Java, y = (m\*x)+c;

Interest = p\*n\*r === In Java

Interest = p\*n\*r;

Flow Control Statements – It controls execution order.

1. Normal flow is line-by-line execution
2. Control statements [if, while, do, for, switch ]

Types of Control Statements

1. Conditional Control Statements (Conditions based)
2. Repetitive/Lopping control Statements [ for, while, do while]