Day4 Revisit

MongoDB

* No SQL, Document based database
* Databases, collections, can be created dynamically
* Social Media, Mobile apps, where the data is highly dynamic (Post)
* Installing MongoDB (Community Version, MongoDBCompass(GUI), MongoAtlas(Cloud DB))
* Mongo
* Use <db\_name> - Eithere create or connect to the database
* Db – will show currently connected database detail
* Show dbs – will display all the available databases
* To store the data, mongodb BSON format – Which is little similar to JSON
* Key can be mentioned with or without quotes in BSON
* In JSON, both key and value should be in double quotes only
* CRUD – insert(), save(),upsert(), update(), remove()
* Client – Server [Request & Response model]
* GUI – Graphical User Interface (MongoCompass) – can connect to both local server & cloud server
* Connection url – for local server - mongodb://127.0.0.1:27017/?readPreference=primary&appname=MongoDB%20Compass&directConnection=true&ssl=false
* Connection url – for remote server – mongo Atlas mongodb+srv://root:root@cluster0.hfeko.mongodb.net/test?authSource=admin&replicaSet=atlas-k8p8rw-shard-0&readPreference=primary&appname=MongoDB%20Compass&ssl=true

Core JAVA

1. Flavors/Editions of JAVA – JAVA SE, JAVA EE, JAVA ME
2. JDK, JRE, JVM
3. JAVA CLASS file – ByteCode – Intermediate Code
4. JDK1.8 (Installed JDK 1.8 from oracle site)
5. Added JAVA\_HOME env variable & updated path to use jdk1.8
6. Keywords, data types (primitive, derived), operators, symbols, statements
7. Writing our first HelloWorld.java and executing it.
8. Downloaded & Installed Eclipse EE IDE (Integrated Development Environment)
9. Access & Non-Access Modifiers (private, package/default, protected, public—final, static, abstract, transient)
10. Conditional Control Statements, Looping Control Statements
11. Class & Object
12. In Java, Everything is Object.

OOP – Generic Concept – It can be implemented in any programming Lang.

OOP – Object Oriented Programming [Data is essential part & it’s very secured]

POP – Procedure Oriented Programming [C] – Importance will be given to the structure and not to the data. Data is not highly secured.

OOP – Class & Object

JAVA is a Object Oriented Programming Lang but it’s not pure OOP lang.

The reason is Java Primitive data types are non-objects.

JAVA is a platform independent programming lang but JDK, JRE & JVM are all platform dependent.

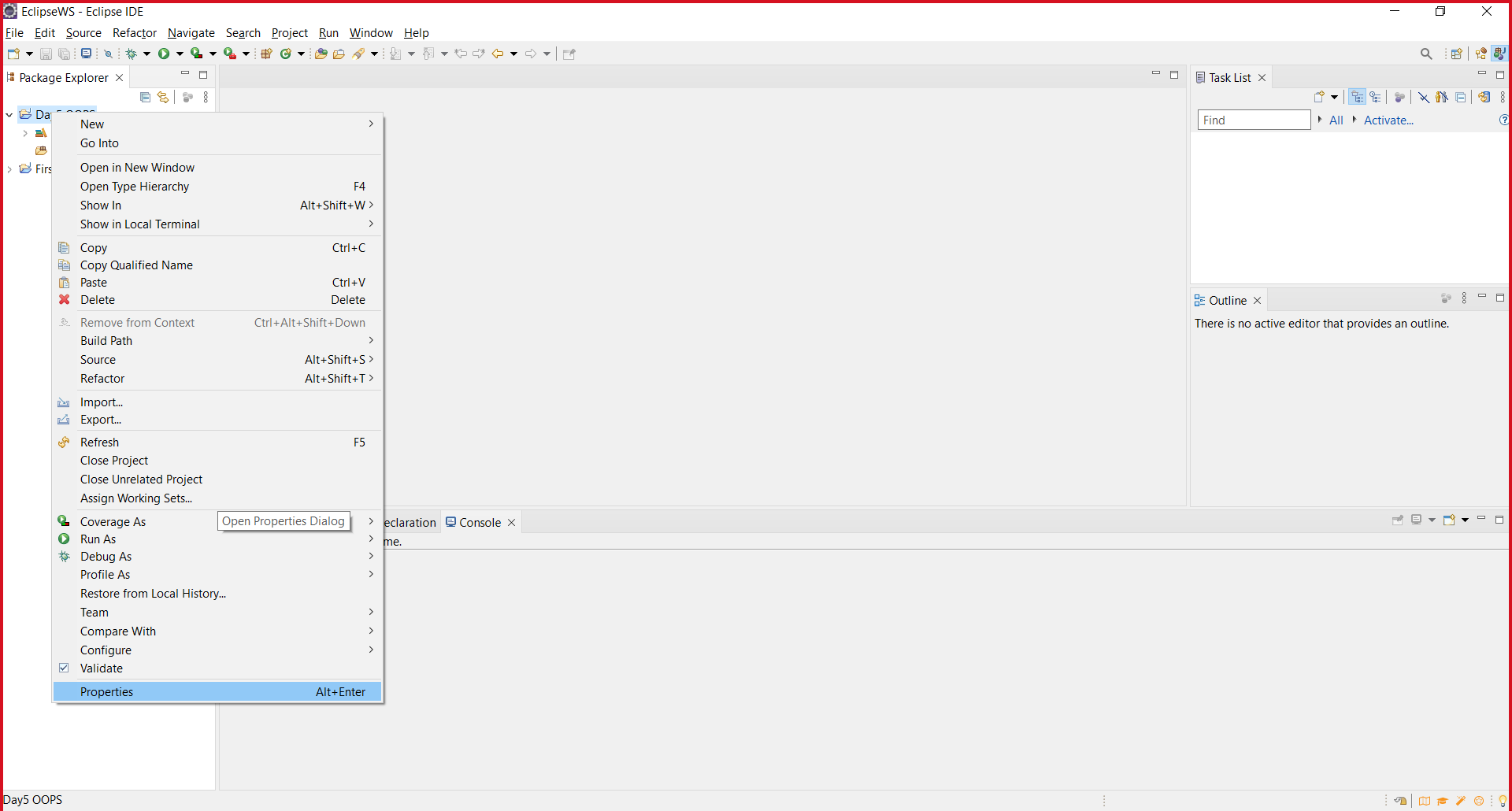
In JAVA Pointers are not allowed -- In JAVA explicit pointer manipulation is not allowed.

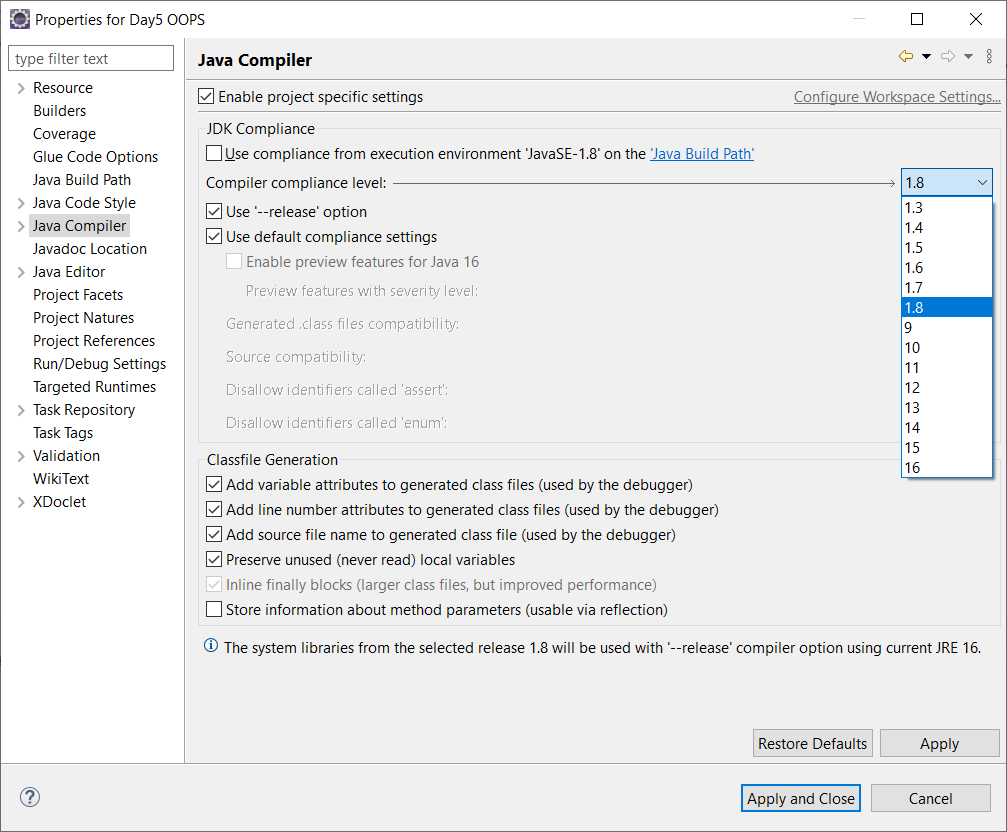
JAVA supports implicit Pointer manipulation. (Reference variable – pointer)

Pointer variable is nothing but a variable which is storing the memory address of another variable.

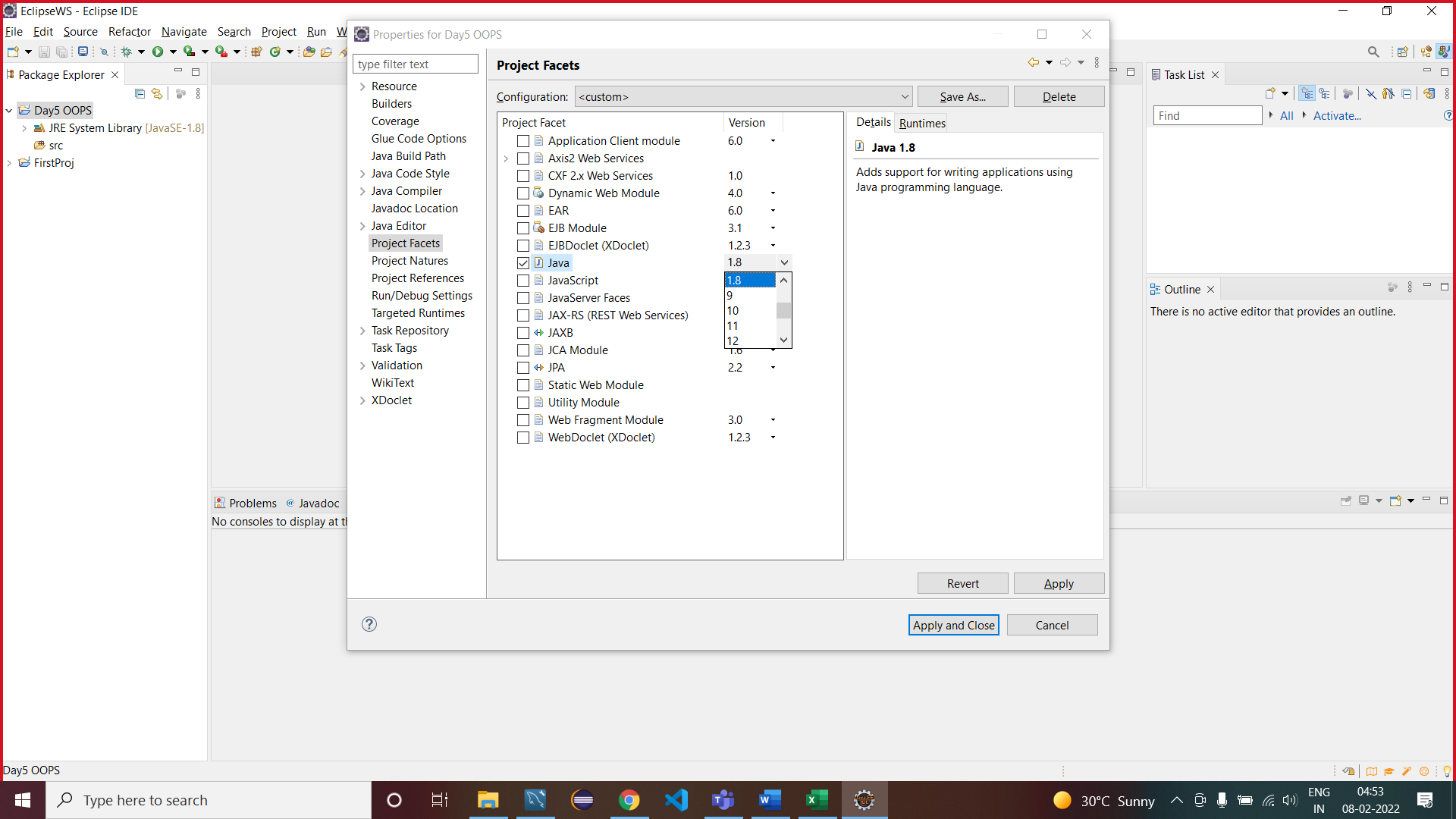
After Creating a Java Project, make sure to check the compiler compilence option (1.8)

Right click the project -> properties -> compiler compilence





Check project-facets JAVA version. Make sure it is pointing to java 1.8



Best Practice for package naming – reverse of the company url – Zensar.com 🡪 com.zensar

Package name should be in lower case only.

Class name should start with CapitalLetter only. If it is more than one word then beginning of each word should in upper case. 🡪 Camel Casing

JAVA follows camel casing convention.

By default, for all the user-defined/custom classes the super/parent class is Object.

In Java, Everything is object.

System.out.println – System is the Built-in Class (A Class which is developed/defined by the developers) – System is a final class which is defined in java.lang package.

Java.lang – package is the default package.

Java Documentation -- JVM will automatically import this package – rt.jar (runtime.jar)

Java.util , java.io - frequently used built-in packages

JDBC – java.sql

Custom package/User defined package – packages defined by the programmers

The name of the public class & the name of the java file should be Same.

In a Java source code file, we can define n number of classes. But only one public class only allowed.

Types Of Classes

1. Simple Class / Concrete Class / Complete Class
2. Abstract Class / In-Complete Class/ Non-concrete Class
3. Starter Class – A class with the main method – This type of class can be run as “Java Application”
4. POJO Class – Plain Old Java Object – A Class which is not extending any other class nor implementing any interfaces.
5. Non-Starter Class – Class without main method.
6. Bean Class – A class with properties, constructors & getter , setter methods
7. Entity Bean Class – A Bean class which represent a table in the Database.
8. Wrapper Classes - (8 wrapper classes –Helps to convert primitive type to its corresponding object type)
9. Final Class – A class with Final access modifier which can’t be extended (can’t create child class)
10. Base Class/ Parent Class/ Super Class
11. Sub Class/ Child Class/ Derived Class
12. Inner Class – A class defined inside another class
13. Anonymous Class – A class without a name

Java is a Case Sensitive Language.

What is Object?

What is Object Reference / reference variable?

Variable declaration, variable initialization.

// - Single Line comment – compiler will ignore that particular line

/\* - Multi line comment start

\*/ - Multi line comment end

//Constructor - Default - No Argument constructor

//Constructor is a special method which will not return any values and it wont have void keyword

//Constructor - To Initialize all the member variable/ properties/ state

//If not defined, jvm will automatically add the default/no-arg constructor.

// public Employee() {

//

// }

//Default / No-Argument Constructor

**public** Employee() {

**super**();

}

//Parameterized -Fully & Partially

//fully parameterized / All Argument Constructor

**public** Employee(**int** id, String name, String email, **long** mobile) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.email = email;

**this**.mobile = mobile;

}

Super – this – keywords

Super refers to parent class

This refers to current object.

This() – The current class constructor

Super() – Parent class constructor

Declaring a variable

Initializing a variable

Ctrl + Shift + F = Shortcut to format your code

Ctrl + Space = Short cut – for auto code completion/ code suggestion

Constructor will initialize all the member variables . It used during object creation.

When initializing the default values, if it is boolean = false , numeric data =0/0.0, char = ‘\0\ (null character), String/Objects null value,

Suppose we need to modify the object’s properties after it’s creation.

We can Getters(), Setter() method for that purpose

Getters() – Accessors – will help you to read/retrieve the value of object’s property

Setters() – Mutators – Will help to set the values to that particular object’s property

Public void set<T>(T t) {

This.t = t;

}

Public T get(T t){

Return t;

}

T – Template variable – int, float, double, String, Object - Generic