Java Basics

1.Class

DEFINITION/WHAT

Is a blueprint/template that tells how Object should be created?

Memory is not allocated when class is created.

NEED/WHY

- ✓ Need 1:An object cannot be created without class declaration
- ✓ Need 2: It contains attributes(data members/instance variables) and behaviour(methods) that an objects constructed from class can exhibit

IMPLEMENTATION/HOW IT WORKS

A class is created using class keyword

REAL TIME EXAMPLE

Generally, when we want to persist some data in database using OR Mapping, we store data in Objects, which require class definition.

2.Object

DEFINITION/WHAT

Is an instance of a class.Memory is allocated when Object is created for a class

NEED/WHY

- ✓ Need 1:To achieve Object orientation
- ✓ Need 2: Class declaration alone is not useful until Object is created

IMPLEMENTATION/HOW IT WORKS

An Object is created when JVM encounters new Keyword

REAL TIME EXAMPLE

Laptop is a class with RAM memory, SSD, keypad as attributes and HP, Lenovo, Dell are Objects of Laptop

ADDITIONAL INFORMATION

Multiple Objects can be created for Single Class.

Memory is allocated depending on size of attributes (datatypes) of the Object.

3. Primitive Datatypes

DEFINITION/WHAT

Datatypes define the size in which memory has to be allocated for a variable. Java has 8 datatypes

DataType	Memory allocated
int	4 Bytes
long	8 Bytes
float	4 Bytes
char	2 Bytes
double	8 Bytes
boolean	1 bit/JVM dependent
short	2 Bytes
byte	1 Byte

NEED/WHY

- ✓ Need 1:size of object is based on total size of the datatype of the data members

REAL TIME EXAMPLE

A student has the attributes roll no, marks etc.,

roll no is of type int and marks is of type float. Thus, 8 bytes of memory is allocated for the student object.

ADDITIONAL INFORMATION

Java has Non Primitive Data Types like String, Array etc.

4. Access Specifiers/Modifier in Java

DEFINITION/WHAT

Encapsulation is achieved by protecting the data members.

Access specifiers specify the visibility of the variable in a class/Package.

Access specifier	**Scope**
private	within the same class
default	Anywhere within the same package
Protected	Anywhere within the same package + child class in another package
public	Anywhere

By default, all data members/instance variables are private and all member functions/methods are public.

NEED /WHY

- ✓ Need 1:In establishing parent-child relation ship
- ✓ Need 2:To achieve Encapsulation

REAL TIME EXAMPLE

Eg: Facebook App

Visibility of any post only for yourself is private

Visibility of post to you and friends is default

Visibility of post to you, friends and friends of friend is protected

Visible to everyone is public

ADDITIONAL INFORMATION

A class can have either default/public access specifier only.

Private and Protected access modifiers cannot be assigned

5. Static Keyword(Non-Access Modifier)

DEFINITION/WHAT

- The static modifier is used along with methods and variables.
- The keyword static indicates that the particular member belongs to a class, and not tagged to the object.

NEED/WHY

- ✓ Need 1: When only one copy of the variable should exist and can be shared by all instances of a class, we can use static keyword

REAL TIME EXAMPLE

Students of a particular College have different name and id, but they share same College Name.

Collage Name can be declared static here, as it is common for all instance of the student.

Additional Information

1. Static variables are initialized when class is loaded.
2. Static variables are initialized before any object of that class is created.

5. Final Keyword (Non-Access Modifier)

DEFINITION/WHAT

- The value of final variable cannot be modified once it is declared. It acts as a constant.
- Final Method cannot be overridden.
- Final Class cannot be extended

REAL TIME EXAMPLE

- `PI = 3.141592653589793`
- `Integer, Float, String`

6. Abstract Keyword (Non-Access Modifier)

DEFINITION/WHAT

- Abstract keyword is used to achieve abstraction.
- Abstract class contains both abstract and non-abstract methods.
- Methods declared with abstract keyword does not contain any implementation.

NEED/WHY

✓ Need 1: Abstract class will provide the default implementation for the common functionalities, while the subclasses can provide the implementation for specific functionalities.

REAL TIME EXAMPLE

ADDITIONAL INFORMATION

A retail firm accepts both online orders and store orders.

Order validation logic is different for each of the channel (store/online).validateOrder() method will be declared as abstract and the implementation is provided in the respective subclasses.

Abstract class - Order Class

Subclass – Online Order

Subclass – StoreOrder