Java class file extension .java

Java is statically typed

Data types -> primitives and non-primitives (wrapper class and reference class)

Booleans are lower case

Auto-boxing is primitive values is automatically converted to Wrapper class

Int j = 100; Integer a = j;

Unboxing is taking primitives out of wrapper class (when we assign object to primitive)

Integer j = new Integer(10); int I = j;

Float into double is widening or up casting

Double to float is narrowing or down casting and may cause data loss

Int[] arr = new int[]; -> wrong! Needs to define array length [<length>]

Int arr[]; arr = new int[5]; -> is 1 correct way

Method(); -> method must be created in a class or part of class members contain block of code

Can be called or invoked by name; named block of code

Method declaration must have: name, return type, access modifier (public, private, protected, default),

parameter declaration (may be empty), ie public void main(String[] args) {body is not manditory}

access modifier -> defines the visibility of the method/variable -> accessibility and scope of a field,

method, constructor, or class

public -> accessible by all in the same project

private -> visible only to class

protected -> visible to the class itself and all subclasses

constructor is a method of a class that instantiates an object of the class, called by new <Object>()

do-while runs and then checks condition to see about running again

class declaration keywords extends, implements, abstract

static keyword -> shared by all instances of the object/class; belongs to the class not the instance

abstraction is a “shows” only essential attributes and “hides” unnecessary information

can have constructor but cannot be instantiated

interfaces cannot have constructor because it can only have abstract methods; also used to achieve abstraction and multiple inheritance; implements keyword; all methods abstract; all fields public final static by default

**encapsulation** – process that binds together the data and functions that manipulate the data that keeps both safe from outside interference and misuse; uses private and getters/setters

**inheritance** – the ability by which one class acquires the properties of another class; uses extends keyword; cannot inherit from multiple class

**overloading** – process of defining more than one method in a class differentiated by method signature or different parameters

**polymorphism** – refers to the ability of a variable, function, or object to take on multiple forms