Interview Questions

Java

What is a class?

Classes are fundamental or basic unit in Object Oriented Programming .A class is kind of blueprint or template for objects. Class defines variables, methods. A class tells what type of objects we are creating. For example take Department class tells us we can create department type objects. We can create any number of department objects.

```
public class FirstClass {
public static void main(String[] args){
System.out.println(“My First class”);
}
}
```

What is an object?

An Object is instance of class. A class defines type of object. Each object belongs to some class.Every object contains state(fields) and behavior. State is determined by value of attributes and behavior is called method. Objects are also called as an instance. To instantiate the class we declare with the class type.

```
public classFirstClass {
public static void main(String[] args) {
 FirstClass f=new FirstClass();
System.out.println(“My First class”);
}
 }
```

To instantiate the FirstClass we use this statement FirstClass f=new FirstClass(); f is used to refer FirstClass object.

What is method in Java?

It contains the executable body that can be applied to the specific object of the class. Method includes method name, parameters or arguments and return type and a body of executable code.

```
Syntax : type methodName(Argument List){

}
```

ex : public float add(int a, int b, int c) methods can have multiple arguments. Separate with commas when we have multiple arguments.

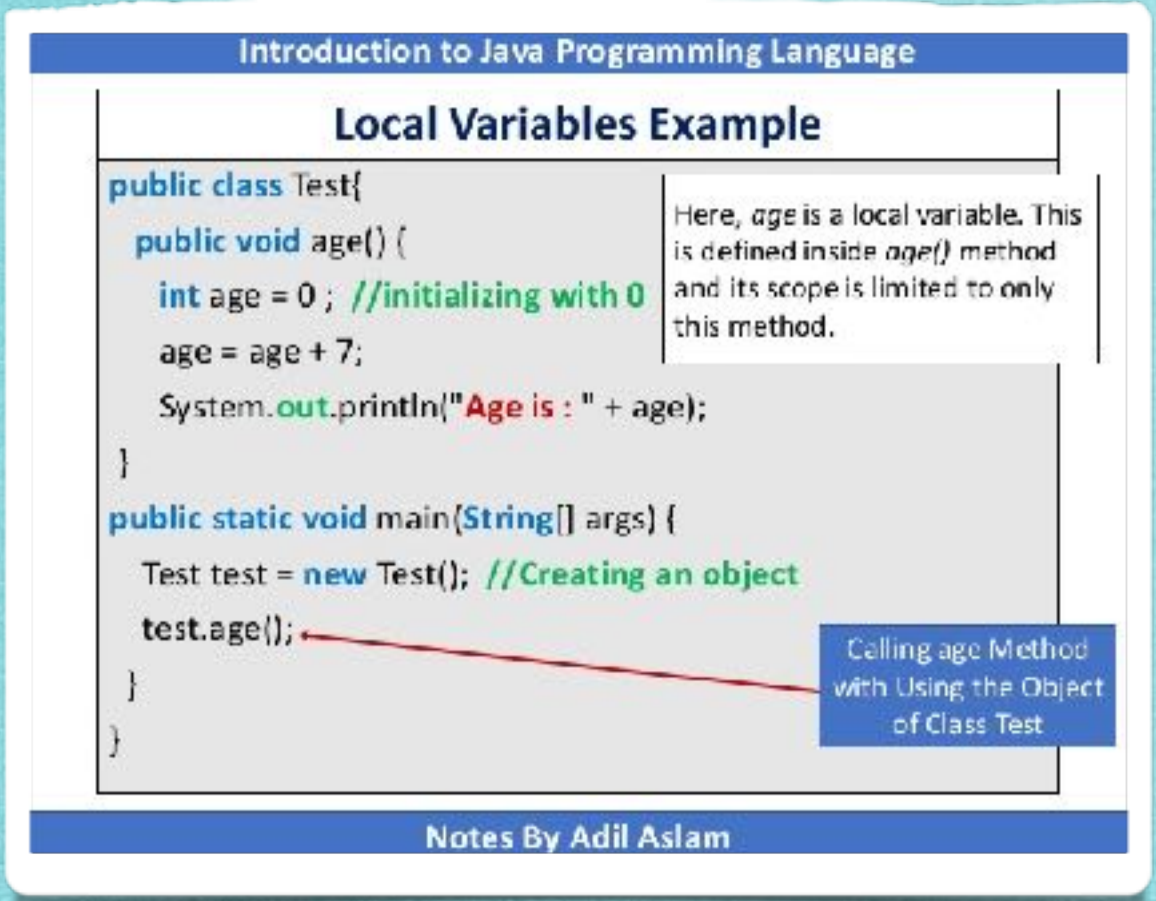
What is a
local variable?
Instance variable?
Class variable?

- Local Variable:Variables defined inside methods, constructors or blocks are called local variables. The variable will be declared and initialized within the method and it will be destroyed when the method has completed.

- Instance Variable:Instance variables are variables within a class but outside any method. These variables are instantiated when the class is loaded.

- Class Variable:These are variables declared with in a class, outside any method, with the static keyword.

Local Variables



Introduction to Java Programming Language

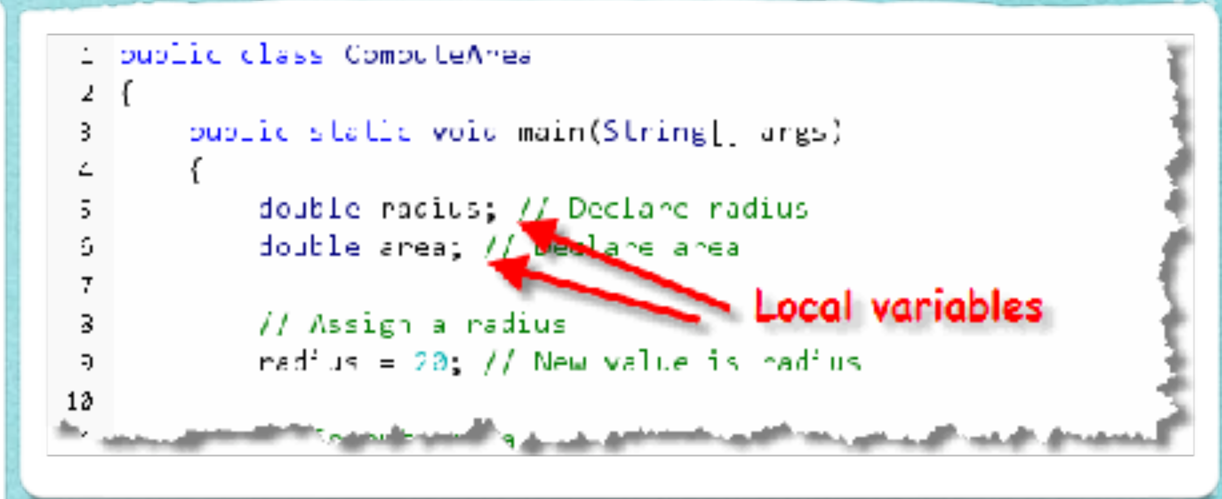
Local Variables Example

```
public class Test{
  public void age() {
    int age = 0 ;  //initializing with 0
    age = age + 7;
    System.out.println("Age is : " + age);
 }
public static void main(String[] args) {
   Test test = new Test();  //Creating an object
   test.age();
 }
}
```

Here, *age* is a local variable. This is defined inside *age()* method and its scope is limited to only this method.

Calling age Method with Using the Object of Class Test

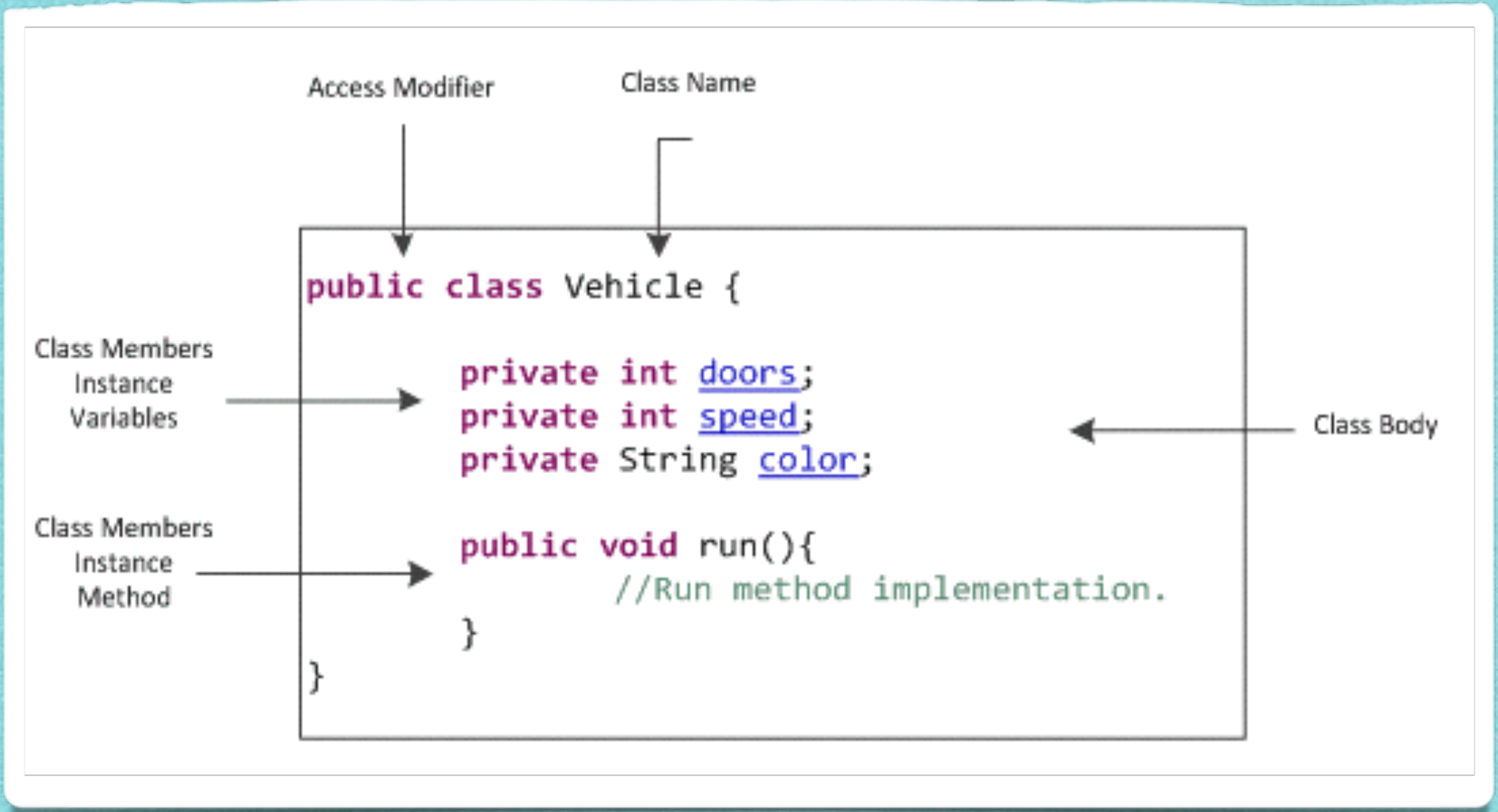
Notes By Adil Aslam



```
public class ComputeArea
{
    public static void main(String[] args)
    {
        double radius; // Declare radius
        double area; // Declare area

        // Assign a radius
        radius = 20; // New value is radius

```

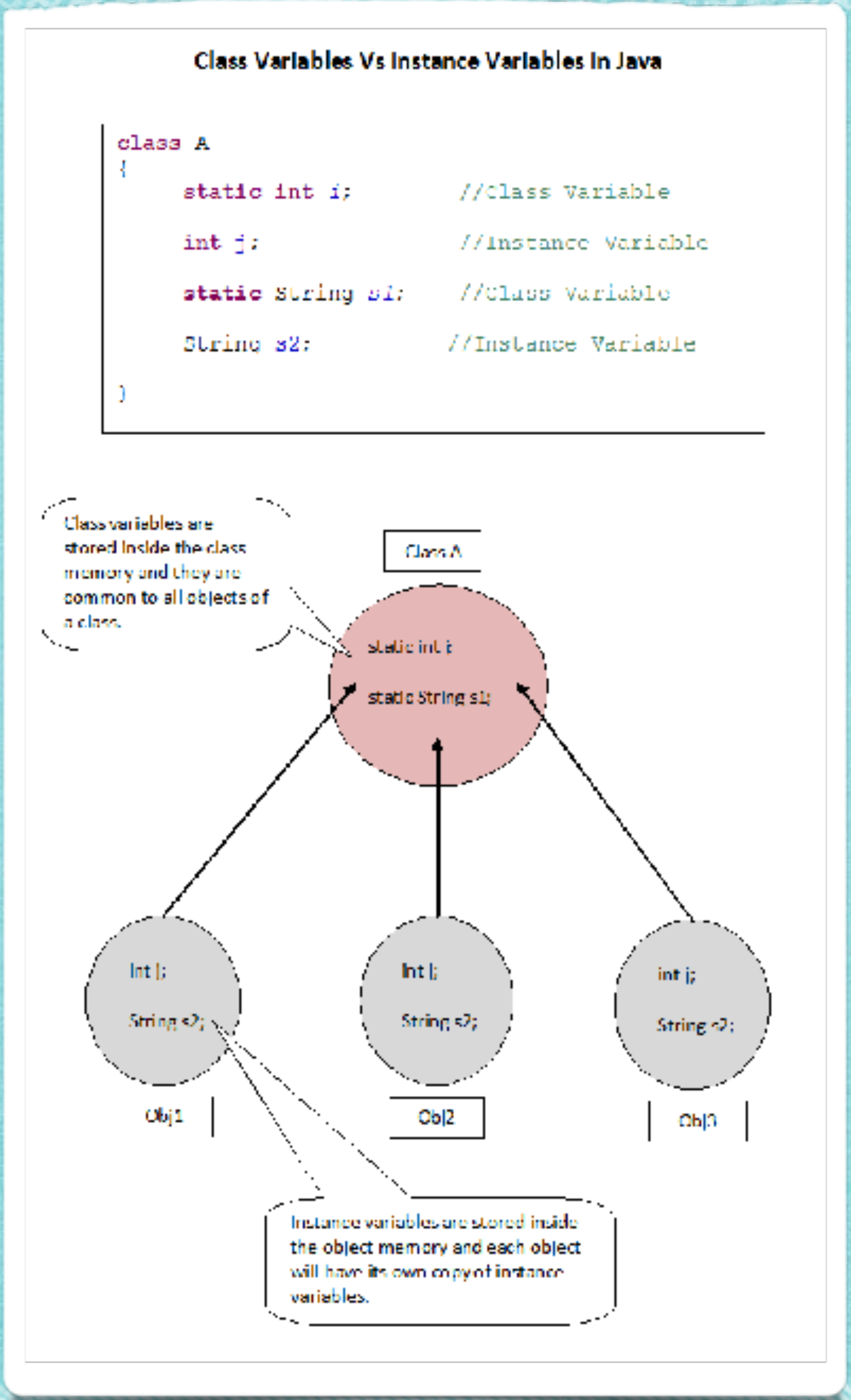
Local variables

Instance Variables



```
public class Vehicle {

        private int doors;
        private int speed;
        private String color;

        public void run(){
                //Run method implementation.
        }
}
```

Access Modifier
Class Name
Class Members Instance Variables
Class Members Instance Method
Class Body



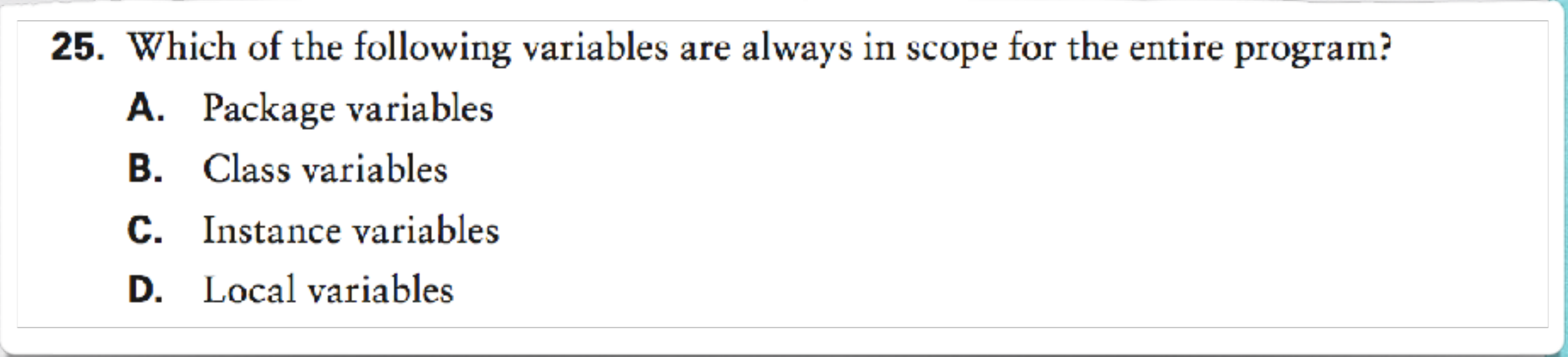
Class Variables Vs Instance Variables In Java

```
class A
{
     static int i;          //Class Variable

     int j;                 //Instance Variable

     static String s1;      //Class Variable

     String s2;            //Instance Variable

}
```

Class variables are stored inside the class memory and they are common to all objects of a class.

Class A
static int i;
static String s1;
Int j;
String s2;
Obj1
Int j;
String s2;
Obj2
int j;
String s2;
Obj3

Instance variables are stored inside the object memory and each object will have its own copy of instance variables.



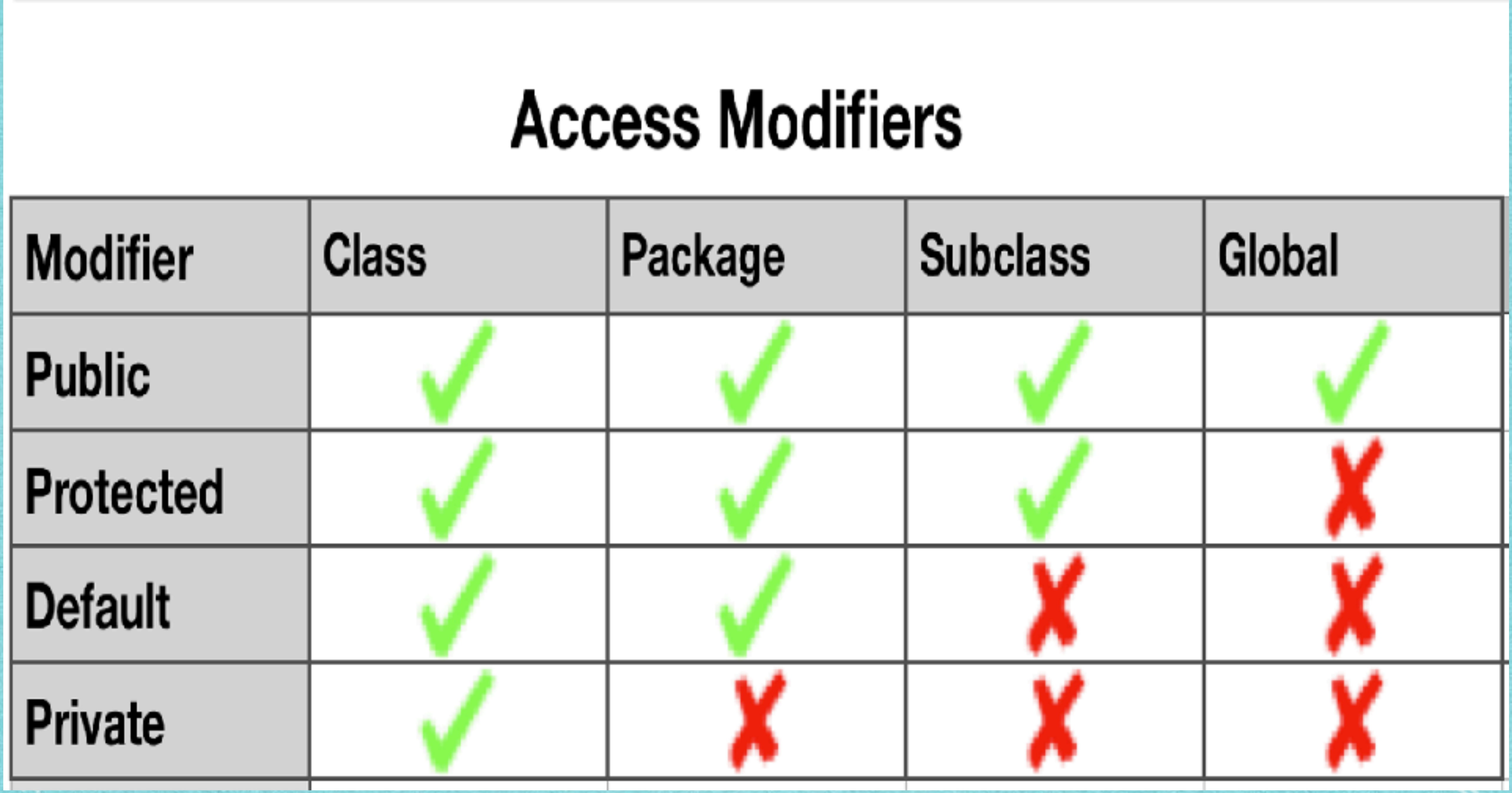
25. Which of the following variables are always in scope for the entire program?

A. Package variables

B. Class variables

C. Instance variables

D. Local variables



Access Modifiers

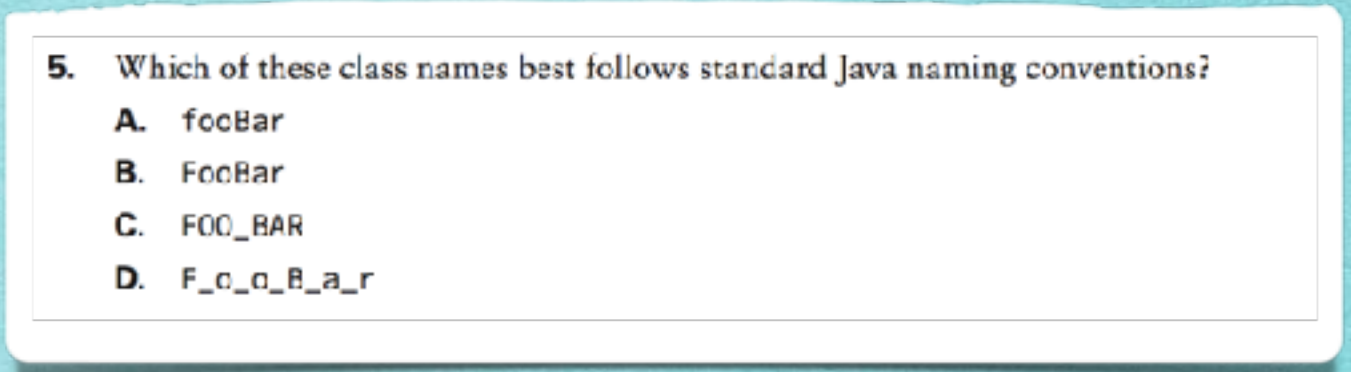
Modifier	Class	Package	Subclass	Global
Public	✓	✓	✓	✓
Protected	✓	✓	✓	✗
Default	✓	✓	✗	✗
Private	✓	✗	✗	✗

What is difference between length and length() method in java?

length() : In String class we have length() method which is used to return the number of characters in string.
Ex : String str = “Hello World”;
System.out.println(str.length());
Str.length() will return 11 characters including space.
length : we have length instance variable in arrays which will return the number of values or objects in array.
For example : String days[]={” Sun”,”Mon”,”wed”,”thu”,”fri”,”sat”}; Will return 6 since the number of values in days array is 6.

What are identifiers in JAVA?

Identifiers are names in java program. Identifiers can be class name, method name or variable name. Rules for defining identifiers in java:
1) Identifiers must start with letter,Underscore or dollar($) sign.
2) Identifiers can't start with numbers .
3) There is no limit on number of characters in identifier but not recommended to have more than 15 characters
4) Java identifiers are case sensitive.
5) First letter can be alphabet ,or underscore and dollar sign. From second letter we can have numbers .
6) We shouldn't use reserve words for identifiers in java.



5. Which of these class names best follows standard Java naming conventions?

A. fooBar

B. FooBar

C. FOO_BAR

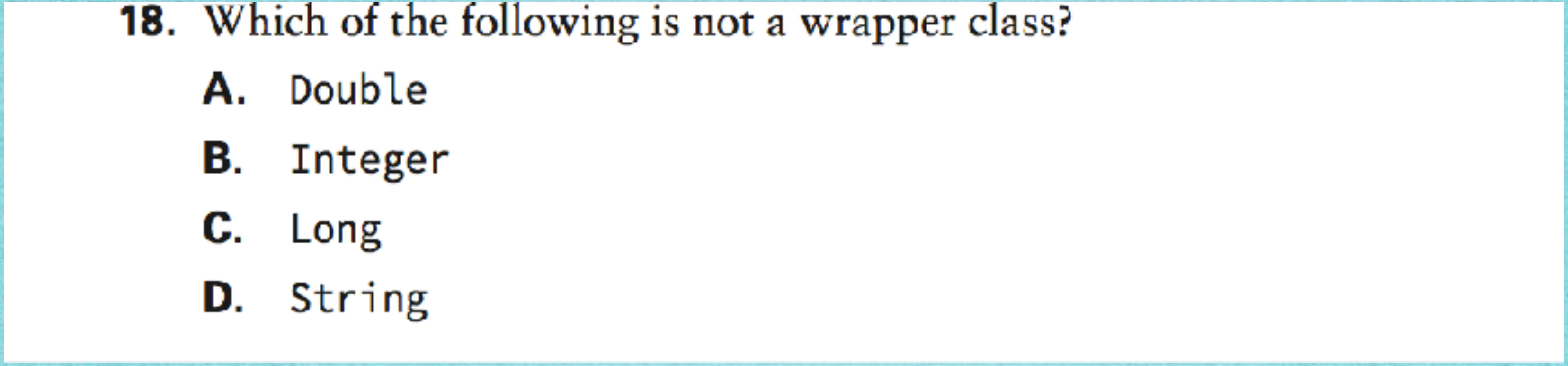
D. F_o_o_B_a_r

How can we explain the main method?

public static void main(String[] args)

public : Public is an access modifier, which is used to specify who can access this method. Public means that this Method will be accessible by any Class.
static : It is a keyword in java which identifies it is class based i.e it can be accessed without creating the instance of a Class.
void : It is the return type of the method. Void defines the method which will not return any value.
main: It is the name of the method which is searched by JVM as a starting point for an application with a particular signature only. It is the method where the main execution occurs.
String args[] : It is the parameter passed to the main method.

What are wrapper classes?



18. Which of the following is not a wrapper class?

- **A.** `Double`
- **B.** `Integer`
- **C.** `Long`
- **D.** `String`

Wrapper classes converts the java primitives into the reference types (objects). Every primitive data type has a class dedicated to it. These are known as wrapper classes because they “wrap” the primitive data type into an object of that class. Refer to the below image which displays different primitive type, wrapper class and constructor argument.

Wrapper Classes

Primitive	Wrapper Class
boolean	Boolean
char	Character
int	Integer
float	Float
double	Double
long	Long
short	Short

What is the difference between equals() and ==?

equal() vs ==

Equals() method is defined in Object class in Java and used for checking equality of two objects defined by business logic.

“==” or equality operator in Java is a binary operator provided by Java programming language and used to compare primitives and objects. *public boolean equals(Object o)* is the method provided by the Object class. The default implementation uses == operator to compare two objects.

For example: method can be overridden like String class. equals() method is used to compare the values of two objects.

What is “IS-A” relationship in JAVA?

‘is a’ relationship is also known as inheritance. We can implement ‘is a’ relationship or inheritance in java using extends keyword. The advantage or inheritance or is a relationship is reusability of code instead of duplicating the code.
Ex : Motor cycle is a vehicle
Car is a vehicle
Both car and motorcycle extends vehicle.

What is "HAS-A" relationship in JAVA?

‘Has a ‘ relationship is also known as “composition or Aggregation”. As in inheritance we have ‘extends’ keyword we don’t have any keyword to implement ‘Has a’ relationship in java. The main advantage of ‘Has-A‘ relationship in java code reusability.

Ex : Car is a vehicle.

Ex : Car has an engine. We cannot say Car is an engine.

What are the difference between Array and ArrayList?

ARRAY	ARRAYLIST
Fixed size and size can not be changed	Dynamic,resizable,and size elements can be modified.It can increase and decrease.

Arrays can accept both primitives and Objects.
double[] d={23.54,23.1 };
String[] s={“One”, “Five”};
	ArrayList can only accept OBJECTS.
	We will need to use wrapper classes to add primitive types.
	ArrayList<Double>nums=new ArrayList<>();

Arrays can be multi dimensional. String[][][] data=new String[5][3][2];	ArrayList cannot be multi dimentional.ArrayList is single dimensional only.
Length	size()
Array is a part of core Java programming and has special syntax. int[] nums={12,34,5}; Print (num[1])—-> print 34	ArrayList is a part of Collection Framework and implements List Interface. ArrayList has methods to read, add, remove etc.

Overloading vs.Overriding

Overloading———Overriding

Same method name different parameters	Same method name same parameters
Occurs in the same class	Occurs in different related classes
CAN overload static, final and private methods	CAN NOT override static,final and private methods
Return type can be same or different	Return type Must be same or co-variant

Constructor() vs. Method(){}

CONSTRUCTOR()	METHOD(){}
Java provides a default constructor(if user did not create one)	Java doesn't provide a method
Constructor name must be same as class name	Name can be same as class name or different
Has no RETURN type	Has return
Constructors invoked implicitly	Methods invoked explicitly
Can't be inherited by a child/sub class	Can be inherited by child/sub class
Called automatically when a new object is created	——-
Can't be private	Can be private

How to call one constructor from the other constructor ?

With in the same class if we want to call one constructor from other we use this() method. Based on the number of parameters we pass appropriate this() method is called.

Restrictions for using this method :

1) this must be the first statement in the constructor

2)we cannot use two this() methods in the constructor

What is super keyword in java ?

super();

Variables and methods of super class can be overridden in subclass .
In case of overriding , a subclass object call its own variables and methods.
Subclass cannot access the variables and methods of superclass because the overridden variables or methods hides the methods and variables of super class. But still java provide way to access super class members even if its members are overridden. Super is used to ac superclass variables, methods, constructors.

Super can be used in two forms :

1) First form is for calling super class constructor.
2) 2) Second one is to call super class variables,methods.
3) Super if present must be the first statement.

Abstract class vs. Interface

ABSTRACT CLASS	INTERFACE
A class can extend only one abstract class.Multiple inheritance is not possible using abstract class.	Any class can implement multiple interfaces.Multiple inheritance is possible using interfaces in java.
Can have constructor	Cannot have constructor
Abstract class can also have private, protected fields.	Fields in Interface are public static final
Abstract class may contain both abstract and concrete methods.	Up to java 7 interface can only contain public abstract methods,We cannot declare any concrete methods inside interface.

Methods in Interface

- 1.abstract methods
  By default in the method signature there is always ‘public abstract’
- 2.default methods
  It’s one way of having the regular method with the implementation in it.Access modifier for default method is public.
- 3.static methods
  It’s also another way of having the regular method with the implementation.And it belongs to Interface.We can call this method using Interface name.
- When we implement Interface in concrete class we MUST implement all unimplemented methods that are inherited.
- Interface cannot implement another interface BUT Interface can extend to another Interface.Also multiple inheritance is allow for Interfaces.