ASSIGNMENT-6

1. What is private access specifier?

Private: The access level of a private modifier is only within the class. It cannot be accessed from outside the class. Private access specifier allows a class to hide its member variables and member functions from other functions and objects. Only functions of the same class can access its private members. Even an instance of a class cannot access its private members.

2. what are getter and setter methods? why do we need them?

Getters and setters are used to protect your data, particularly when creating classes. For each instance variable, a getter method returns its value while a setter method sets or updates its value. Given this, getters and setters are also known as accessors and mutators, respectively.

3. why this keyword in the setter method?

For example, in the following code:

```
private int id;

public void setlD(int ID) {
   this.id = ID;


public void getID() {
   return id;
```

In setters, we have a method argument with the same name as the instance variable . In the code, one ID and one id, but we could have called them both id. Then this. would be required to distinguish the instance variable from the method argument

4.Difference between local variable and member variable/instance variable.

Local variable	Instance variable
A variable that is used inside the method.	A variable that is inside the class but outside the method.
It is not possibl[l]e to use access modifiers	It is possible to use access modifiers
Do not have default values	Can have default values
Local variables create when entering a method or constructor	Instance variables create when creating an object

5.What is reference variable?

A reference variable is a variable that points to an object of a given class, letting you access the value of an object. An object is a compound data structure that holds values that you can manipulate. A reference variable does not store its own values.

6. **Syntax of creating an object?**

The syntax for creating an object is:

ClassName object — new ClassName();

7. **Explain in detail what happens when we create an object?**

When an object is created, memory is allocated to hold the object properties. An object reference pointing to that memory location is also created. To use the object in the future, that object reference has to be stored as a local variable or as an object member variable.

8.What is class?

Class is a template used to create objects and to define object data types and methods. Classes are categories, and objects are items within each category. All class objects should have the basic class properties.

9.What is object?

object is a member (also called an instance) of a Java class. It be created to call a non-static function which are not present inside the Main Method but present inside the Class and also provide the name to the space which is being used to store the data.

10. What are the default values of all the datatypes?

Data Type	Default Value (for fields)
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
String (or any object)	null
boolean	false

11.Difference between the static methods and instance method?

Sr. NO	Instance Methods	Static (Class) Methods
1	Instance methods are used to perform repetitive tasks like reading records from the file, reading records from the DBMS etc.-	Static methods are used to perform single operation Like opening the files, obtaining a DBMS connection etc.-.
2	Methods definition does not require a static keyword to start. Syntax: void net_salary @arameters if any) Block of statements;	Methods definition must start with the static keyword. Syntax: static void basic_salary (parameters if any) Block of statements;
3	Each and every instance method must be accessed with the respective Object name	Each and every static method must be accessed with the respective Class name
4	Result of instance method is not shared. Every object has its own copy of instance method.	Results of static method always shared by objects of the same class.

12.Syntax of accessing the member variable in the main?

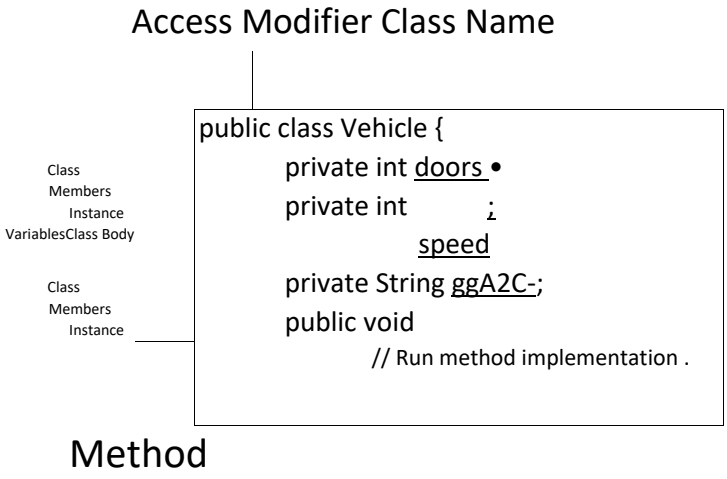
Syntax: ObjectReference.VariableName.

13. Syntax of instance method definition?

Syntax: Access _specifier void method_name( )

Statement;

Example:

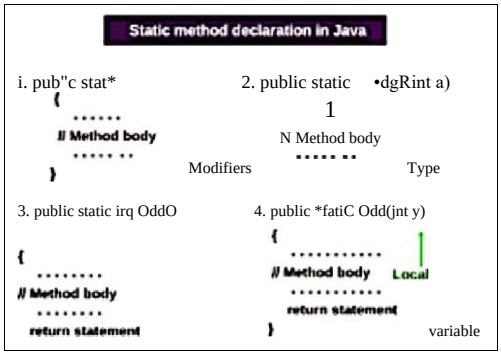


14.Syntax of static method definition?

Syntax: Access_modifier static void methodName()

// Method body.

Example:



15.Difference between actual parameter and formal parameter?

Formal arameter	Actual arameter
Formal parameter is the parameter defined in the function header.	Actual parameter is the argument passed to the function durin the function call.
Syntax: // function header type func name (type pa ram 1 , type param2 // function body	Syntax: // function call statement func name (paraml, param2, ;
Example: int sum (int a, int b) return a+b; Here, paraml and param2 are the formal arameters of sum åmction.	Example: sum (x, Y) ; Here, x and y are the actual parameters passed to the sum ftnction and gets mapped to the formal parameters paraml and param2 respectively.
The formal parameters are locally available within the particular firnction only.	The actual parameters are part of the calling function. Example: If main method calls a function sum. then main is the calling function and sum is the called method.

16. Why we need the parameter or arguments to the methods?

An argument is a value passed to a function when the function is called. Whenever any function is called during the execution of the program there are some values passed with the function

A parameter is a variable used to define a particular value during a function definition. Whenever we define a function we introduce our compiler with some variables that are being used in the running of that function.