1. What is private access specifier?

- **The access level of a private modifier is only within the class**.
- It cannot be accessed from outside the class.
- The access level of a default modifier is only within the package.
- It cannot be accessed from outside the package.
- If you do not specify any access level, it will be the default.

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?

- The get method returns the value of the variable name .
- The set method takes a parameter (newName) and assigns it to the name variable.
- The this keyword is **used to refer to the current object**.

4. difference between localvariable and member variable/instance variable.

- **A member variable is a member of a type and belongs to that type's state.**
- **A local variable is not a member of a type and represents local storage rather than the state of an instance of a given type**.

5. what is reference varialble?

- 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 you create an object, **you are creating an instance of a class, therefore "instantiating" a class**.
- The new operator requires a single, postfix argument: a call to a constructor.
- The name of the constructor provides the name of the class to instantiate.
- The constructor initializes the new object.

8. what is class?

- ★ **Class is a blueprint which defines some properties and behaviors.**
- ★ A class is not allocated memory when it is defined.
- ★ Class is a logical entity whereas objects are physical entities.

9. what is object?

- object is **a member (also called an instance) of a Java class**.
- Each object has an identity, a behavior and a state.
- The state of an object is stored in fields (variables), while methods (functions) display the object's behavior.
- Objects are created at runtime from templates, which are also known as classes.

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?

Static:

- **static methods exist as a single copy for a class. instance methods exist as multiple copies depending on the number.**
- Static methods can't access instance methods.
- static variables and static methods directly.

Instance:

- **instance methods exist as multiple copies depending on the number.**
- **instances created for that particular class**.
- variables directly while instance methods can access. static variables and static methods directly.

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

<datatype> <variable_name>; <datatype> <variable_name> = <initializing_value>;

13. Syntax of instance method defination?

- An instance method is a method that belongs to instances of a class, not to the class itself.
- To define an instance method, just **omit static from the method heading**.
- Within the method definition, you refer to variables and methods in the class by their names, without a dot.

14. syntax of static method defination?

- Static methods are created in Java using the static keyword.
- The syntax for creating a static method is as follows: **Java**. **public static void methodName() { // method body }**

15. difference between actual parameter and formal parameter?

- ★ **Actual parameters are those parameters that are specified in the calling function.**
- ★ **formal parameters are those parameters that are declared in the called function**.

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

Parameters refers to the list of variables in a method declaration. Arguments are the actual values that are passed in when the method is invoked. **When you invoke a method, the arguments used must match the declaration's parameters in type and order**.

17. why we need the return statement and return type to the method.

A return statement **causes the program control to transfer back to the caller of a method**. Every method in Java is declared with a return type and it is mandatory for all java methods. A return type may be a primitive type like int, float, double, a reference type or void type(returns nothing).

18. Method can be private.(true or false)

A private method is an access modifier used in a class that can only be called from inside the class where it is defined. It means that you cannot access or call the methods defined under private class from outside.

19. what is the error message if we access private variable or method out side the class?

- **You cannot override a private or static method in Java**.

- If you create a similar method with same return type and same method arguments in child class then it will hide the super class method
- this is known as method hiding.
- Similarly, you cannot override a private method in sub class because it's not accessible there.