# OOPS

# Fundamentals Assignment

1. How to create an object in Java?

Ans. The object is a basic building block of an OOPs language. In Java, we cannot execute any program without creating an object.Using the new keyword is the most popular way to create an object or instance of the class. When we create an instance of the class by using the new keyword, it allocates memory (heap) for the newly created object and also returns the reference of that object to that memory. The new keyword is also used to create an array. The syntax for creating an object is:

```
ClassName object = new ClassName();
```

2. What is the use of a new keyword in Java?

Ans. The new keyword in Java is used to create an instance of a class, also known as an object. The new keyword in Java is used to allocate memory for the object on the heap, the memory space where objects are stored. The new keyword in Java calls the constructor of a class to initialize the object's state.

3. What are the different types of variables in java?

Ans. A variable is a container which holds the value while the Java program is executed. A variable is assigned with a data type.

Variable is a name of memory location. There are three types of variables in java: local, instance and static.

There are two types of data types in Java: primitive and non-primitive.

1) Local Variable

A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists.

A local variable cannot be defined with "static" keyword.

2) Instance Variable

A variable declared inside the class but outside the body of the method, is called an instance variable. It is not declared as static.

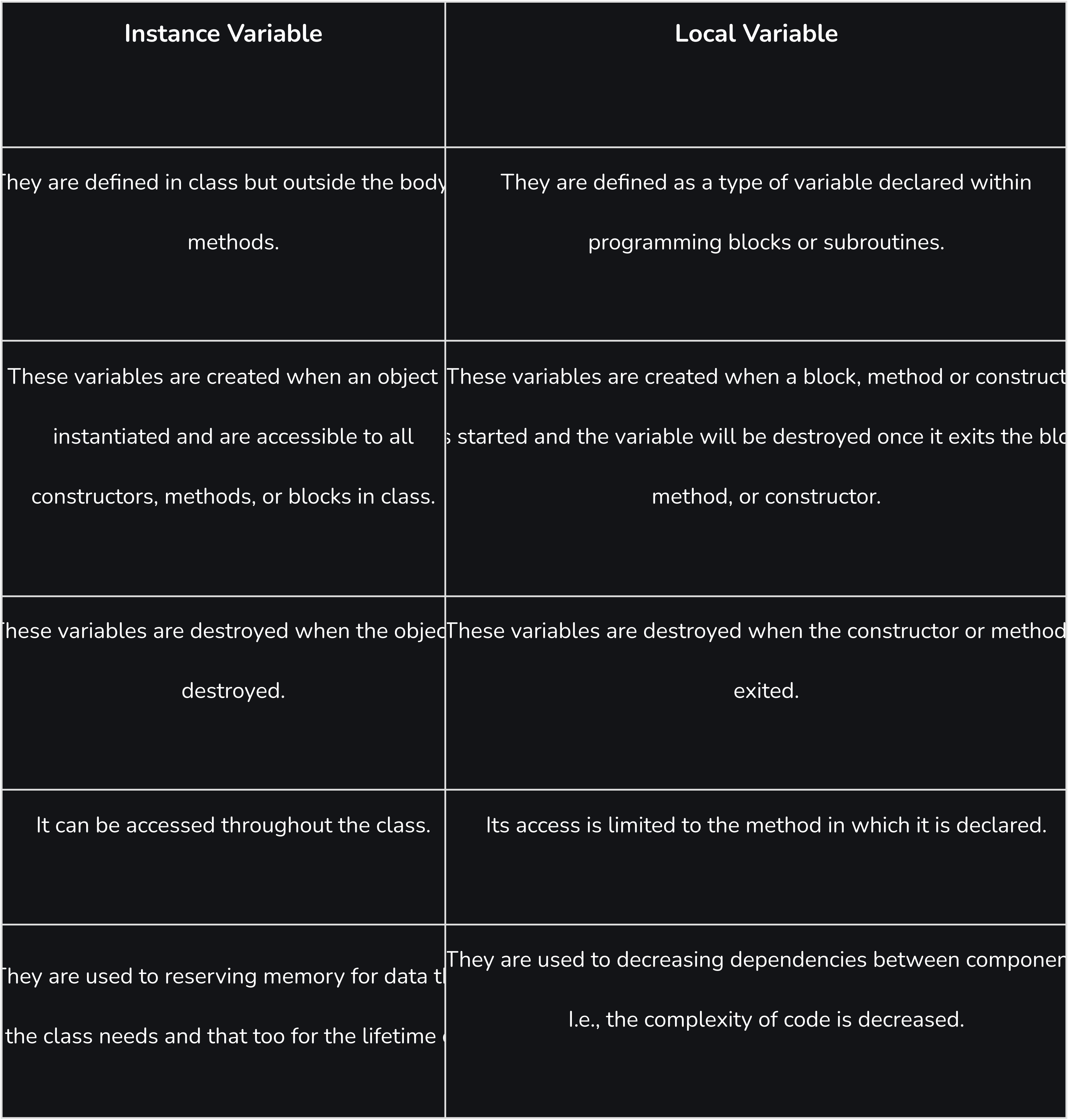
It is called an instance variable because its value is instance-specific and is not shared among instances.

3) Static variable

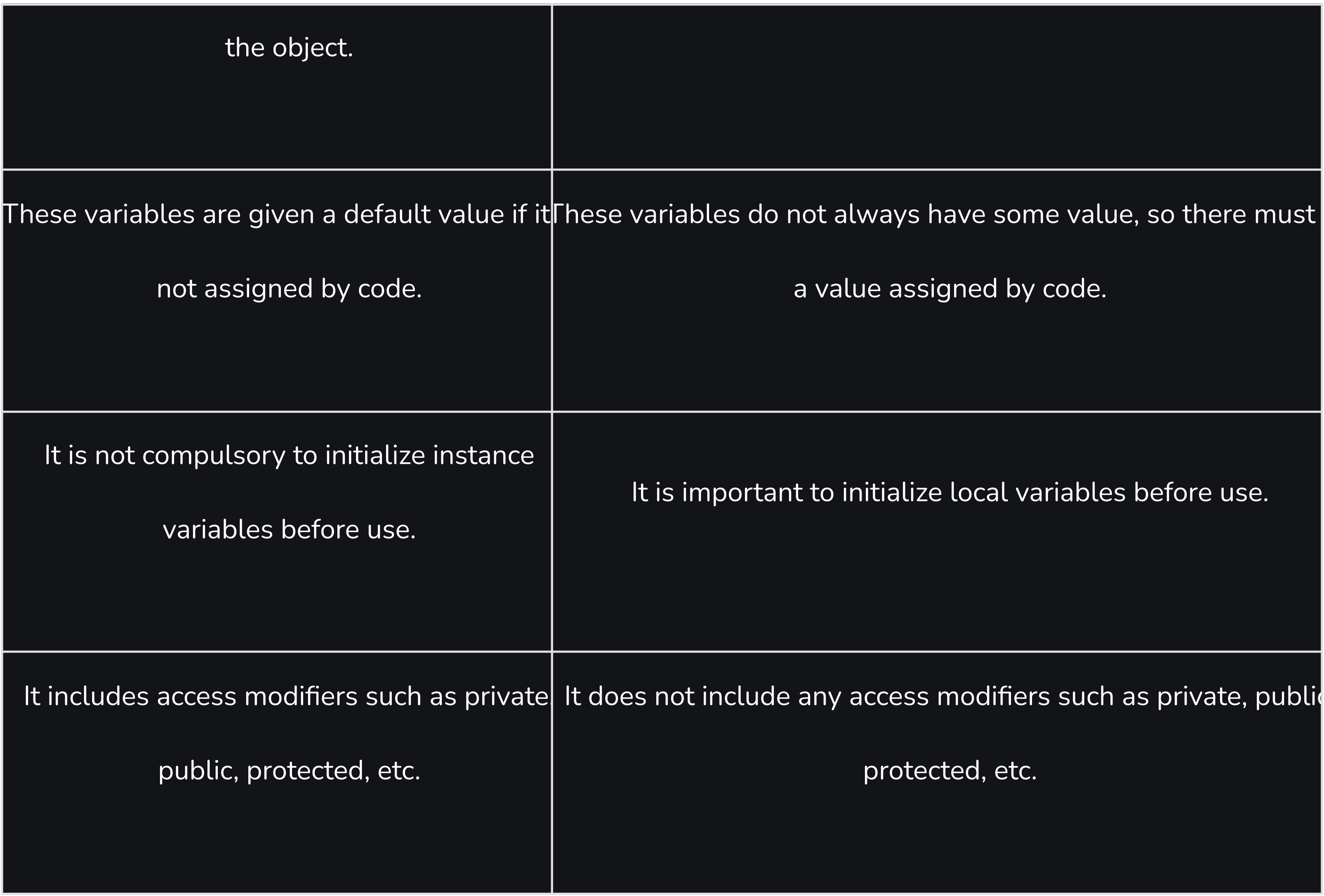
A variable that is declared as static is called a static variable. It cannot be local. You can create a single copy of the static variable and share it among all the instances of the class. Memory allocation for static variables happens only once when the class is loaded in the memory.

4. What is the difference between instance variable and local variable?

Ans.



| Instance Variable | Local Variable |
|---|---|
| They are defined in class but outside the body methods. | They are defined as a type of variable declared within programming blocks or subroutines. |
| These variables are created when an object instantiated and are accessible to all constructors, methods, or blocks in class. | These variables are created when a block, method or construct s started and the variable will be destroyed once it exits the blo method, or constructor. |
| These variables are destroyed when the objec destroyed. | These variables are destroyed when the constructor or method exited. |
| It can be accessed throughout the class. | Its access is limited to the method in which it is declared. |
| They are used to reserving memory for data t the class needs and that too for the lifetime | They are used to decreasing dependencies between componen I.e., the complexity of code is decreased. |



| | |
|---|---|
| the object. | |
| These variables are given a default value if it not assigned by code. | These variables do not always have some value, so there must a value assigned by code. |
| It is not compulsory to initialize instance variables before use. | It is important to initialize local variables before use. |
| It includes access modifiers such as private, public, protected, etc. | It does not include any access modifiers such as private, public protected, etc. |

5. In which area memory is allocated for instance variable and local variable?

Ans. memory for instance variables and local variables is allocated in different areas:

Instance Variables:

- Memory for instance variables is allocated on the heap.
- Each instance of a class (object) has its own set of instance variables, and these variables are created when an object is instantiated.
- Instance variables exist throughout the lifetime of the object to which they belong.

Local Variables:

  - Memory for local variables is allocated on the stack.
  - Local variables are declared within a method, constructor, or block of code and are only accessible within that scope.
  - They are created when the method is called or the block of code is entered and are destroyed when the method exits or the block of code completes.

- Instance variables belong to an object and are stored on the heap.
- Local variables are temporary and are stored on the stack.

6. What is method overloading?

Ans. Method Overloading allows different methods to have the same name, but different signatures where the signature can differ by the number of input parameters or type of input parameters, or a mixture of both.

Method Overloading in Java is also known as compile-time Polymorphism.

```
public class Sum {

public int sum(int x, int y) { return (x + y);

}

   public int sum(int x, int y, int z)

 {

   return (x + y + z);

}

 public double sum(double x, double y)

 {

  return (x + y)
```

```
   }

  public static void main(String args[])

   {

   Sum s = new Sum();

  System.out.println(s.sum(10, 20));

   System.out.println(s.sum(10, 20, 30));

   System.out.println(s.sum(10.5, 20.5));

 }

}
```

Output:-

30

60

31.0