Oops Fundamentals

Assignment-

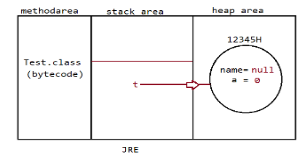
Question 1- How to create an object in Java?

Ans

To create an object in Java, we need to follow the following steps:

1.We can use the “new” operator to create a new object.

2.There is no “delete” operator in java because destruction of useless objects is the responsibility of the garbage collector.



Question 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. When you use the ‘new’

Keyword followed by the name of a class, Java creates a new object of that class and

returns a reference to it.

For example, consider the following code:



Here, we’re creating a new object of the’MyClass’ class using the ‘new’ keyword. The variable ‘myobject’ now refers to the newly created object.

Using the ‘new’ keyword, we can create multiple instances of the same class, each with their own set of instance variables and methods. This allows us to create complex programs that manipulate many objects at the same time.

It’s important to note that when you create a new object using the ‘new’ keyword, the object is allocated memory on the heap. Once the object is no longer needed, it’s important to free up this memory by setting the reference to ‘null’.

Question 3- What are the different types of variables in Java?

Ans-

1. Local variables: These are variables declared inside a method or block of code. They are only accessible within that method or block of code.
2. Instance Variables: These are variables declared within a class but outside of any method.They are also known as fields or attributes. They are accessible by all methods in the class and can be accessed by creating an object of the class.
3. Static Variables: These are variables declared with static keywords and are also known as class variables. They are associated with the class rather than any particular instance of the class. They are initialized with the class rather than any particular instance of the class. They are initialized only once when the class is loaded and are accessible by all instances of the class.

Question -4 What is the difference between instance variable and Local variables?

Ans-

In Java, an instance variable is a variable that is defined at the class level and is associated with each instance of the class. On the other hand, a local variable is a variable that is defined within a method or block and is only accessible within that method or block.

Somekey that differences between instance variables and local variables in Java:

1. Scope: Instance variables have a wider scope than local variables. Instance variables can be accessed by any method or constructor within the same class, whereas local variables can only be accessed within the method or block where they are defined.
2. Lifetime: Instance variables exist as long as the object exist , whereas local variables have a shorter lifetime and are destroyed once the method or block they are defined in completes its execution.
3. Access modifiers: Instance variables can have access modifiers,public, private or protected, whereas local variables cannot have access modifiers.
4. Initialization: Instance variables are initialized to default values when an object is created, whereas local variables must be initialized before they can be used.
5. Memory allocation: Instance variables are allowed memory when an object is created, whereas local variables are allocated memory when the method or block they are defined in is executed.
6. Concurrency: Instance variables can be accessed concurrently by multiple threads if not synchronized properly, whereas local variables are thread-safe since they are only accessible within the method or block they are defined in.

All this means to say that ,instance variables represent the state of an object, whereas local variables are used to hold temporary values or perform intermediate calculations within a method or block.

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

Ans-

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

1. Instance Variables: Memory for instance variables is allocated on the heap, which is a shared area of memory used by all threads in a Java application. When an object is created using the “new” Keyword, memory is allocated on the heap for all its instance variables.
2. Local Variables: Memory for local variables is allocated on the stack, which is a special area of memory used for method execution. When a method is invoked, a new frame is created on the stack to store its local variables and any other necessary data. When the method returns, the frame is removed from the stack, and the memory used by the local variable is freed.

It’s important to note that instance variable are part of an object’s state and persist as

long as the object exists, while local variable are only valid within the scope of the

method in which they are declared.

Question 6- What is method overloading?

Ans-

Method overloading is a feature in Java that allows a class to have multiples with the same

Name but with different parameters. When a method is overloaded, the compiler

Determines which version of the method to cal based on the number, type ,order of the

Parameters passed to the method.

For an example of method overloading in Java:

