30 Jan'23 Oops Fundamentals Assignment - 8

1. How to create an object in java?
Ans: We can use the **"new"** operator to create an **object**.

There is no **"delete"** operator in java because destruction of useless objects is the responsibility of the garbage collector.

Code:

```
class Test
{
String name;
int a;
public static void main(String[] args) {
Test t = new Test();
}
}
```

2. What is the use of a new keyword in java?
Ans: We can use the **"new"** operator to create an **object**.

3. What are the different types of variables in java?
Ans: Based on the behavior and position of declaration all variables are divided into the following 3 types.

1. Instance variables

2. Static variables

3. Local variables

4. What is the difference between instance variables and local variables?
Ans: **Local variables** are **visible** only in the **method or block** they are declared whereas **instance variables** can be seen by all **methods** in the **class**. Place where they are declared: **Local**

variables are declared inside a **method** or a **block** whereas **instance variables** inside a **class**, but outside a **method**.

Instance Variable: These variables are declared within a class but outside a method, constructor, or block and always get a default value.

These variables are usually created when we create an object and are destroyed when the object is destroyed.
We may use an access specifier, for instance, variable, and if no access specifier is specified, then the default access specifier is used.
Each and every object will have its own copy of instance variables.
Example:

```
class Taxes
{
  int count; // Count is an Instance variable
  /*...*/
}
```

Local Variable: These variables are declared within a method but do not get any default value.

They are usually created when we enter a method or constructor and are destroyed after exiting the block or when the call returns from the method.
Its scope is generally limited to a method and its scope starts from the line they are declared. Their scope usually remains there until the closing curly brace of the method comes.
The initialization of the local variable is mandatory.
Example:

```
int area()
{
	int length = 10; // Local variable
	int breadth = 5; // Local variable
```

```
        int rectarea = length*breadth; // Local variable
        return rectarea;
}
```

5. In which area memory is allocated for instance variable and local variable?
Ans: Memory allocation of instance variable:
Objects and corresponding **instance variables** will be stored in the **Heap area**

Memory allocation of local variable:
For every method the JVM will create a Runtime stack, all method calls performed by that Thread and corresponding **local variables** will be stored in that **stack**. Every entry in **stack** is called **Stack Frame** or **Action record**.

6. What is method overloading?
Ans: Method overloading enables several methods to use the same name but have distinct signatures, where the signature might vary based on the quantity, nature, or combination of input arguments.

- In the 'C' language we can't take 2 methods with the same name and different types. If there is a change in argument type compulsory we should go for a new method name.

Example :
abs() for int datatype
labs() for long datatype
fabs() for float datatype

- Lack of overloading in "C" increases complexity of the programming.

- But in java we can take multiple methods with the same name and different argument types.

abs(int) for int datatype
abs(long) for long datatype

30 Jan'23 Oops Fundamentals Assignment - 8

abs(float) for float datatype