Assignment Answers

1. How to Create an Object in Java

Answer:

- The ' new ' operator is used to create an object
- However , there is no " delete " operator in java to delete objects. The destruction of useless objects is the responsibility of the " Garbage Collector. "

For example:

```java
        class ObjectCreation
{
  int n;
  int name;
  public static void main(String[] args) {
      ObjectCreation obj = new ObjectCreation();
  }
}
```

- The operator " new " is a keyword to create an object
- When we say ' new ', the JVM will allocate the memory for that object on the heap memory.
- When we run the .class file after compilation of .java file , JVM will first load the .class file data on " Method Area ".
- JVM will initialize the memory for the instance variable on the heap memory as a part of object at the time of object creation.
- JVM will set the default values for the instance variables based on the respective data types if they don't initialize at the time of declaration.
- Once the memory for the object is set , then the address of the object is stored in the reference variable which is a local variable present in the "Stack Area".

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

Answer: The ' new ' operator is used for the object creation. It creates an object . When we say the keyword " new " then , JVM will start the allocation of the memory for the object on the heap memory.

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

Answer:

Division 1 : Based on the values stored by the variables , all variables are divided into two types . They are

   (a) . Primitive variables : These variables store the primitive values (which means pre-defined type of values broadly data type values like int , float , double , byte , short type values).
Ex : int x = 2;
Here ' x ' is a variable which stores int type value(2).

(b) . Reference variables : These variables are used to refer the Objects.
Ex: Test t = new Test(); // where Test is a class and ' t ' is a referece variable.

Division 2 : Based on the behavior and position of declaration , all variables are divided into 3 types. They are

- Instance variables

- Static variables
- Local variables

4. What is the difference between Instance variable and Local variables?

Answer: Instance Variables :

- The variables whose values are varied from object to object are called Local variables.
- The Instance variables are created at the time of object creation and destroyed at the time of respective object destruction . So , the scope of an object is the exactly same as the scope of the corresponding object.
- The Instance variables are stored on the ' heap ' memory as a part of object.
- JVM will assign the default values for these variables if they don't initialize at the time of declaration.
- Instance variables can be directly access from Instance area . But can't access from the static area
- By using the object reference we can access the instance variables from the static area.
- Instance variables are created inside the class but outside any method.

Local Variables :

- To meet the requirements of the programmer , we create the temporary variables within the method , block or constructors. These are called as Local variables or automatic variables or temporary variables or stack variables
- These variables are stored in the Stack memory
- These may include primitive and reference variables.
- These variables are created at the time of block execution in which they created and destroyed after completion of execution. So , the scope of local variable is exactly same as the scope of the block in which they created.
- JVM won't give any default values for the local variables . They should be exiplicitly initialized at the time of declaration.

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

Answer: For Instance variables , memory is allocated on the Heap memory .

And for the local variable , memory is allocated on the Stack memory.

6. What is method overloading?

Answer: Two methods are said to be overloading if and only if they have same names and different argument types. The methods having same method names but different argument types , then we said method overloading