How are Java objects stored in memory?

Difficulty Level : Basic Last Updated : 12 Jul, 2021

In Java, all objects are dynamically allocated on Heap. This is different from C++ where objects can be allocated memory either on Stack or on Heap. In C++, when we allocate the object using new(), the object is allocated on Heap, otherwise on Stack if not global or static.

In Java, when we only declare a variable of a class type, only a reference is created (memory is not allocated for the object). To allocate memory to an object, we must use new(). So the object is always allocated memory on heap (See this for more details).

For example, following program fails in the compilation. Compiler gives error *"Error here because t is not initialized".*

```
class Test {

// class contents
void show()
{
    System.out.println("Test::show() called");
}
}

public class Main {

        // Driver Code
    public static void main(String[] args)
    {
        Test t;

        // Error here because t
        // is not initialized
        t.show();
    }
}
```

Allocating memory using new() makes above program work.

```
class Test {

// class contents
void show()
{
    System.out.println("Test::show() called");
}
}

public class Main {

    // Driver Code
    public static void main(String[] args)
    {

        // all objects are dynamically
        // allocated
        Test t = new Test();
        t.show(); // No error
    }
}
```

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.