44-542 Object-Oriented Programming

Head and Stack Memory

Heap memory is allocated for program resources at run time and Stack memory is used to hold the method specific local variables and resources immediate after the method execution is completed stack memory is released and available for re use

1.You are given a class file MemoryTest.java and code to create an object and test the memory functionality of the memory allocation and management by java runtime

Understand the below code and answer the following questions

public static void main(String[] args) {

Object testMemory = new Object();

MemoryTest memoryTestObject = new MemoryTest();

memoryTestObject.memoryUnderstanding(memoryTestObject);

}

public class MemoryTest {

public String memoryUnderstanding(Object o) { // Line 6

String retValue = "";

return retValue = o.toString();

}

}

1.When you run the program for the first time what method is called and which memory is created and allocated to the method?

Solution

Main method and Stack memory

2.When the line of code Object testMemory = new Object(); is executed where the Object is stored and where the object variable testMemory is stored? What did you understand form this?

Solution

Object is stored on Heap memory and testMemory is stored on Stack memory. Whenever we use a new keyword the object created will be stored in Heap memory. And, the reference variable is stored on Heap and object variable is stored on Stack memory

3.When you call the method memoryUnderstanding() what memory is allocated and why?

Solution

All the method resources and methods are allocated stack memory so when program invokes the method memoryUnderstanding() it allocated to new memory in the stack.

4.Where does the retValue is stored ?

Solution

Stack memory

5.When does the stack memory allocated for method resources is released ?

Solution

Immediate after method returns a value or method execution is terminated the stack memory is released and available for reuse.

6. Write the line of code from the above program after which Question 6 solution justified

Solution

return retValue = o.toString();

7.What do you understand by Garbage Collection and explain user intervention to achieve this for Stack and Heap memory?

Solution

Garbage Collection is collecting the unused memory and clearing it so that Java can use the memory space for future resources. In heap memory Java automatically provides the garbage collection and clears the unused memory .For stack memory user has to explicitly terminate the method by returning a value or exiting the method so that the stack memory is cleared and available for reuse.