Assignment 1

Name – Shashi

Date - 12/28/2017

1. What is javac and java?

Java c is command to start the java compiler. it compiles source code into a program that can be run. If you program's name is "Test1" then (javac Test1.java)

Java command starts a java application.

2. When objects are distracted from heap area?

heap allocations are done with the new keyword. you can free them with the delete keyword each new must have its delete counterpart for the specified object and type. If not the object are destroyed from heap area.

3. Gather information about java stack and pc register?

   Java Stack memory is used for execution of a thread. They contain method specific values that are short-lived and references to other objects in the heap that are getting referred from the method.
   Stack memory is always referenced in LIFO (Last-In-First-Out) order. Whenever a method is invoked, a new block is created in the stack memory for the method to hold local primitive values and reference to other objects in the method.
   As soon as method ends, the block becomes unused and become available for next method.
   Stack memory size is very less compared to Heap memory.

   Pc register - The Java Virtual Machine can support many threads of execution at once. Each Java Virtual Machine thread has its own pc (program counter) register. At any point, each Java Virtual Machine thread is executing the code of a single method, namely the current method for that thread. If that method is not native, the pc register contains the address of the Java Virtual Machine instruction currently being executed. If the method currently being executed by the thread is native, the value of the Java Virtual Machine's pc register is undefined. The Java Virtual Machine's pc register is wide enough to hold a return Address or a native pointer on the specific platform.

4. Why oracle maintain interrupter though we have JIT compiler?

   Though both compiler and an interpreter do the same job of converting a high-level language to a machine executable code, there are few differences in the way they do it.

   A JITcompiler is a program that converts the entire code into a equivalent machine code at once. That is a compiler scans the entire program first, translate it to a machine executable code which will then be executed by the process and the corresponding output is generated. Compiler generates errors if any at the end of execution of the program.

On the other side, **interpreter** takes the single instruction of the code, translates it into an intermediate code and then into a machine code, executes it and takes another instruction. Interpreter generates errors as soon as it finds error in any of the sequence of instructions.