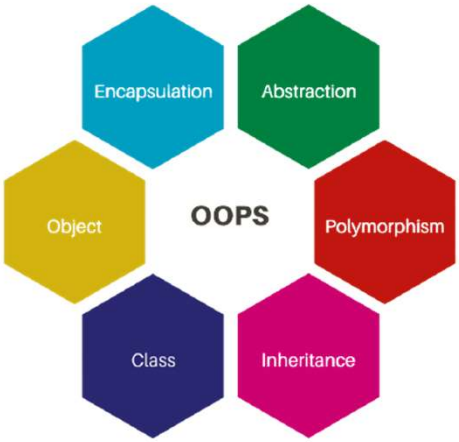


ABESIT COLLEGE OF ENGINEERING
CRC-Training

Java™

Memory Management in Java

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Memory Management in Java

Memory management in Java refers to the process of allocating and freeing up space for objects. Java automatically manages memory.

The memory is logically divided into two primary sections - Stack and Heap.

- All local variables and method invocations are stored in the stack
- All objects along with their instance variables are stored in the heap

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Creation of objects

- Class is a blueprint for the creation of objects
- To realize a class, an object of the class needs to be created
- An object is an instance of a class
- There can be many instances for a class and each instance will have its own data

In Java, the operator ***new*** allocates memory for objects during run time ie. ***dynamic memory allocation***

The following statement creates an object of the class Customer and returns a ***reference*** to the newly created object

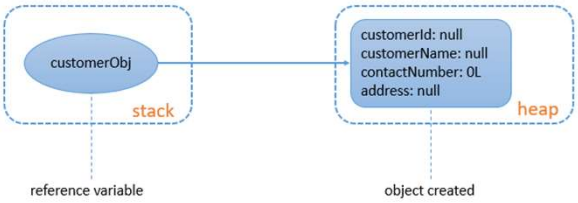
```
new Customer();
```


ABESIT COLLEGE OF ENGINEERING
CRC-Training

Creation of objects

```
class Customer{
        int customerId;
        String customerName;
        long contactNumber;
        String address
        }
class  retail{
        public static void main(String args[]){

        Customer custmerObj = new Customer();
        }
}
```



Please note that reference variables are also local variables. Reference variables are local variables which stores the address of another memory location.

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Reference Variables

The ***reference*** returned by a newly created object must be assigned to a variable

This variable is called **reference variable**

Following syntax can be used to create a reference for the Customer class

```
Customer custObj;
```

Reference variable can be assigned **'null'** to show that it is not referring to any object

null is a keyword

```
Customer custObj = null;
```

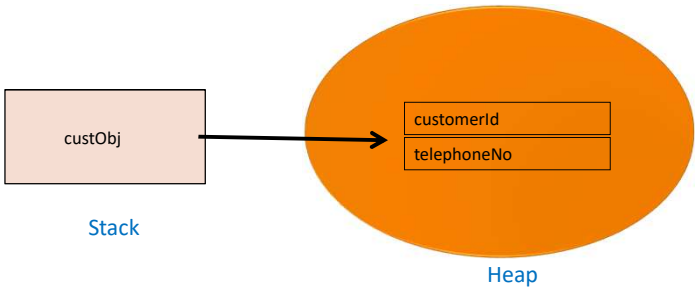
5

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Reference variables & Objects in Memory

Customer custObj = new Customer();

Where will the memory be allocated for the reference variable and object?



6

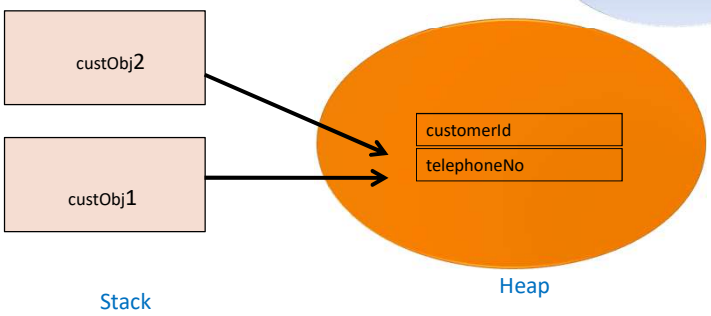
ABESIT COLLEGE OF ENGINEERING
CRC-Training

Reference variables & Objects in Memory

Consider the code given below:

```
Customer  custObj1 = new Customer();
Customer  custObj2 = custObj1;
```

How many objects and reference variables will be created ?



ABESIT COLLEGE OF ENGINEERING
CRC-Training

Garbage Collection

- Automatic garbage collection is the process of looking at heap memory, identifying which objects are in use and which are not, and deleting the unused objects
- An in use object, or a referenced object, means that some part of your program still maintains a pointer to that object
- An unused object, or unreferenced object, is no longer referenced by any part of your program

ABESIT COLLEGE OF ENGINEERING
CRC-Training

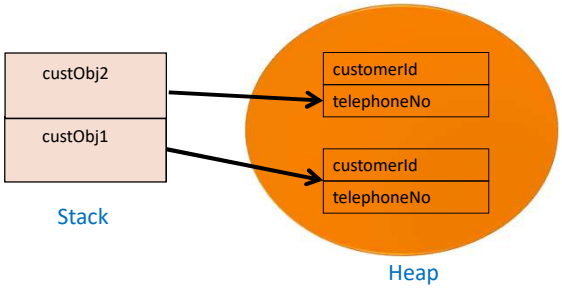
Garbage Collection

- Dynamically allocated memory that is no longer needed should be de-allocated
- In Java, de-allocation is done by a Garbage Collector
- It is a system-level thread to keep track of memory allocations
- Functions of Garbage Collector:
  - Checks for and frees memory no longer needed
  - Is run automatically by the JVM

ABESIT COLLEGE OF ENGINEERING
CRC-Training

References & objects

```
Customer custObj1= new  Customer();
Customer custObj2= new Customer();
```



How many Reference variables?

References : 2

How many Objects?

Objects :2

ABESIT COLLEGE OF ENGINEERING
CRC-Training

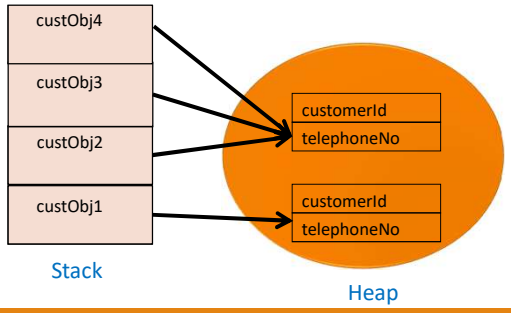
References & objects

Customer custObj1= new Customer();

Customer custObj2= new Customer();

Customer custObj3= custObj2;

Customer custObj4= custObj3;



How many Reference variables?

References : 4

How many Objects?

Objects : 2

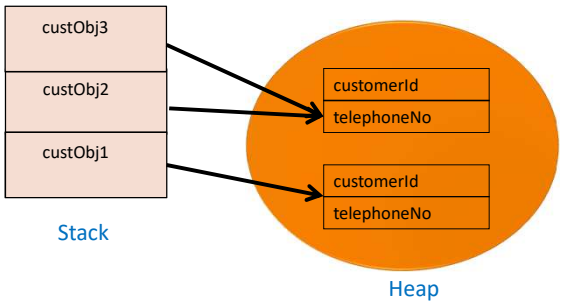
ABESIT COLLEGE OF ENGINEERING
CRC-Training

References & objects

Customer custObj1= new Customer();

Customer custObj2= new Customer();

Customer custObj3= custObj2;



How many Reference variables ?

References :3

How Many Objects?

Objects :2

ABESIT COLLEGE OF ENGINEERING

CRC-Training

Can you answer these questions?

Q1. Consider the below given code snippet:

```
class MyClass{
        public int myVariable;
        public void myMethod(){
                int temp=10;
                ……..
        }
        public static void main(String args[]){
                MyClass obj=new
MyClass();
                ……..
        }
}
```

Ans:
myVariable – Heap
temp – Stack
obj (Reference - local variable) – Stack

Where the following stored in memory? myVariable, temp, obj,

13