Programme	: B. Sc.
Subject	: Computer Science
Semester	: V
Paper Code	: CSC106
Paper Title	: Object Oriented Programming
Unit V	: Memory Management
Module Name	: Garbage Collection, Practical issues of garbage collection, reference counting
Module No	: 15
Presenter	: Ms. Prajoti Chimulkar, Assistant Professor St. Xavier's College, Mapusa - Goa.

What is Garbage Collection?

When an object created in memory is no more required, it must be removed so that memory can be reused for other objects. Removing unwanted objects or abandoned objects from the memory is called garbage collection (GC).

Garbage collection is an essential part of memory management system. In a typical garbage collection cycle all objects that are still referenced, and thus reachable, are kept. The space occupied by previously referenced objects is freed and reclaimed to enable new object allocation.

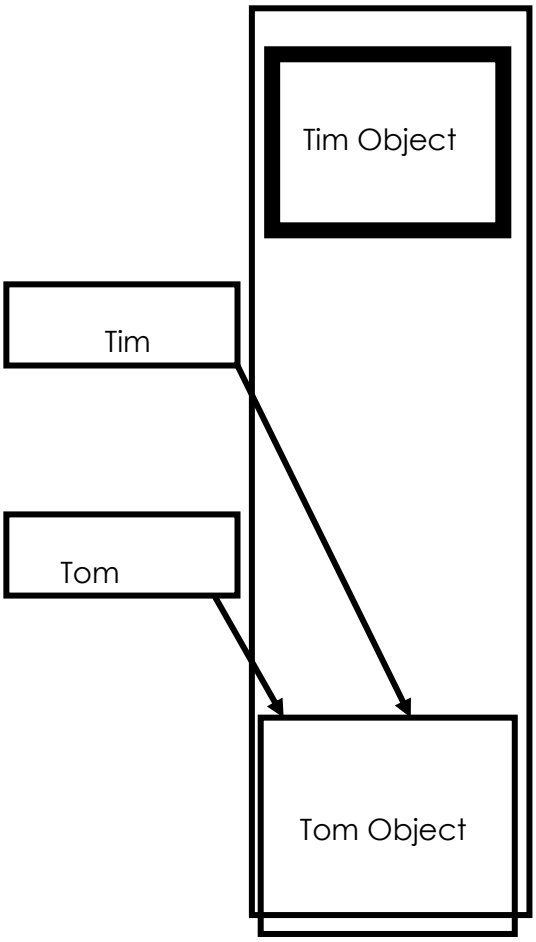
```
Student Tim = new Student();

Student Tom = new Student();

Tim = Tom
```

Now Tim Object becomes a garbage,

It is unreferenced Object



Common Garbage Collection Schemes

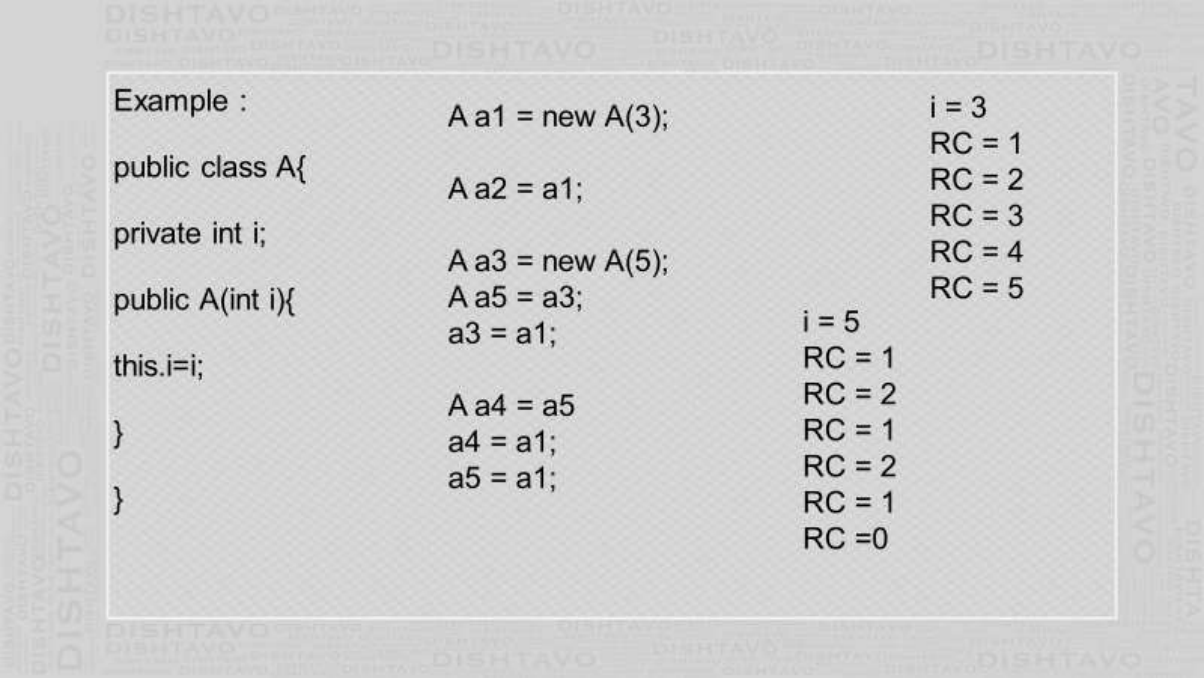
Three main methods of garbage collection:

- ❑ Reference counting
- ❑ Mark-and-sweep
- ❑ Stop-and-copy

Reference Counting Garbage Collection

Main Idea: Add a reference count field for every object. This Field is updated when the number of references to an object changes.

Reference counting collectors keep track of how many references are pointing to each Java object. Once the count for an object becomes zero, the memory can be immediately reclaimed. This immediate access to reclaimed memory is the major advantage of the reference-counting approach to garbage collection.



Mark and Sweep Garbage Collection

Any garbage collection algorithm must perform 2 basic operations. One, it should be able to detect all the unreachable objects and secondly, it must reclaim the heap space used by the garbage objects and make the space available again to the program.

The above operations are performed by Mark and Sweep Algorithm in two phases:

1) Mark phase

2) Sweep phase

The mark-and-sweep algorithm is divided into two phases:

Mark phase: The garbage collector traverses the graph of references from the root nodes and marks each heap object it encounters. Each object has an extra bit: the mark bit – initially the mark bit is 0. It is set to 1 for the reachable objects in the mark phase.

Sweep phase: The garbage collector scans the heap looking for objects with mark bit 0 – these objects have not been visited in the mark phase – they are garbage. Any such object is added to the free list of objects that can be reallocated. The objects with a mark bit 1 have their mark bit reset to 0.

Stop-and-Copy Garbage Collection

In stop-and-copy garbage collection, the heap is divided into two separate regions. At any point in time, all dynamically allocated object instances reside in only one of the two regions--the active region. The other, inactive region is unoccupied. When the memory in the active region is exhausted, the program is suspended and the garbage-collection algorithm is invoked. The stop-and-copy algorithm copies all of the live objects from the active region to the inactive region. After the copying is completed, the active and inactive regions exchange their roles. Since the stop-and-copy algorithm copies only the live objects, the garbage objects are left behind. In effect, the storage occupied by the garbage is reclaimed all at once when the active region becomes inactive.