

Auto garbage collector :gc();

Employee emp = new Employee();                      dynamic memory.

C in language

In using malloc or calloc we can create dynamic memory

**free()** function to destroy

### Object Serialization

Storing the object in external file or converting object in **byte** format is known as object serialization.

Object

Property	id,name,salary
Behaviour	method setId,getId
Identity	reference of object.

We are storing only property not behaviour as well as not identity.

Which class object we are planning to store or we want to do serialization that class must be implements **Serializable interface**.

Serializable interfaces doesn't contains only method or zero.

The interface contains zero method or no method that type of interface is known as **marker interface**.

Object serialization is converting object into byte format.

Object de-serialization : converting byte to object format.

## Collection Framework with Data Structure

Data : in any format primitive format ie number, string, Boolean or reference format ie array, pre defined or user defined class object.

Organization the data in proper way which help to do searching, sorting, storing, retrieve in efficient way.

Data structure help to improve to search the data in proper way.

Time complexity and memory to store the data.

Java provided lot of pre defined classes which internally provided implementation of all data structure.

In java we can create custom data structure API.

Variable

```
int a=10;
```

```
a=20;
```

array : more than one value of same type.

```
int abc[]={10,20,30,40,50}
```

it allow to store only same type of values.

Structure : it is a type of user defined data type which help store

Different types of values

But C or C++, or C# support structure but java doesn't

```
struct Employee {
```

```
    int id;
```

```
    float salary;
```

```
};
```

```
class Employee {
```

```
    int id;
```

```
    String name;
```

```
    float salary;
```

```
}
```

Class advanced of structure.

```
Employee emp = new Employee();
```

```
emp.id=100;
```

```
emp.name="Ravi";
```

```
emp.salary=12000;
```

```
emp.id=101
```

array objects

```
Employee employees[]=new Employee[10];
```

zero object of employee

```
int abc[]=new int[10];
```

class created..

```
employees[0]=new Employee();
```

```
employees[1]=new Employee();
```

```
employees[2]=new Employee();
```

till

```
employees[9]=new Employee();
```

## primitive array or array object limitation

1. They are fixed in memory size.
2. They doesn't provide any pre fined method to add, remove, search the data from array ie primitive or user defined object.
3. Array allow to store same type of values.

## Collection Framework

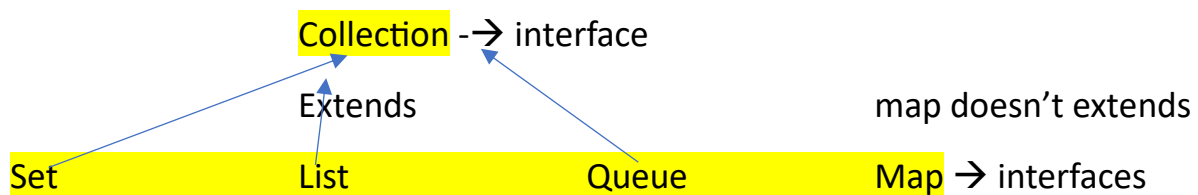
Collection framework provided set of classes and interfaces which help to store any types of values int, float, string, Boolean as well as user defined class object.

They provide dynamic memory features.

The provide lot of pre defined method we can add, remove, search, iterate very easily.

Util package.

### Collection Framework hierarchy



**Set** : it doesn't allow duplicate data. Under set implementation classes may be unordered, ordered or sorted. Set doesn't provide index.

Set implementation classes

1. HashSet
2. LinkedHashSet
3. TreeSet

**List** : it allow duplicate. It maintains the order and provide index features to get the value.

List implementation classes

1. ArrayList
2. LinkedList
3. Vector

#### 4. Stack

**Queue :** Queue is a type of data structure. Queue provide First In First Out. FIFO.

Queue implementation classes

1. PriorityQueue
2. LinkedList

Set, List and Queue we can store one-one information of any type.

**Map :** it allow to store the information in key-value pairs.

Key can be unique and value may be duplicate.

Map implementation classes

1. HashMap
2. LinkedHashMap
3. TreeMap
4. Hashtable