Assignment 9

(1) What is the significance of equals & hashcode methods?

→ The equals() & hashcode() are the two important methods provided by the object class for comparing objects.

(*) Since the object class is the parent class for all java objects, hence all objects inherit the default implement of these two methods.

1) equals() → is used to compare equality of two objects.

Syntax: public boolean equals (Object obj)

2) hashcode() → returns hascode value as an integer. hashcode value is mostly used in hashing based collection like HashMap, HashSet, HashTable....etc. this method must be overridden in every class which overrides equals() method.

Syntax : public int hashcode();

(2) How is hashcode used by hash based collections like HashSet?

→ Collections such as HashMap & HashSet use a hashcode value of an object to determine how it should be stored inside a collection & the hashcode is used again in order to locate the object in its collection.

(Ex) public class GFG {

```
Psvm(...) {
HashSet<Integer> arr = new HashSet<Integer>();
ar.add(1);
ar.add(2);
```

```
al.add(3);
al.add(4);
S.O.P("HashSet:" + arr);
S.O.P("HashCode:" + al.hashCode());
}}
```

O/P: HashSet: [1,2,3,4]
HashCode: 10

(3) what is the Contract b/w hashCode() & equals() ? Why follow it during class design ?

→ The hashCode() method should return the same integer value for the same integer object for each calling of this method unless the value stored in the object is modified.

(4) if two objects are equals (according to equals() method) then the hashCode() method should return the same integer value for both the objects.

(4) Write an Algorithm.

(1) Linear Search : It is used to search a key element from multiple elements. Linear search is less used today bec/ it is slower than binary search & hashing.

Algorithm : (1) traverse the array

(2) match the key element with array element

(3) if key element is found, return the index position of the array element

(4) if key element is not found, return -1

class LinearExample {

public static int LinearSearch (int[] arr, int key) {

for (int i=0; i < arr.length ; i++) {

if (arr[i] == key) {

return i; } }

return -1; }

Public static void main (String a[]) {

int[] a1 = { 10, 20, 30, 40, 50, 70, 90 };

int key = 50 ;

S.O.P (key + " is found at Index " + LinearSearch (a1, key));

② Binary Search : is used to search a key element from multiple elements & it is faster than linear search.

→ In case of Binary Search, array elements must be in ascending order. If you have unsorted array, you can sort the array using Arrays.sort (arr) method.

Ex: class BinaryExample {

public static void binarySearch (int arr[], int first, int last, int key)

{ int mid = (first + last) / 2 ;

while (first <= last) {

if (arr[mid] < key) {

first = mid+1 ; }

else

if (arr[mid] == key) {

S.O.P (" Element found at index " + mid);

break ;

} else

{ last = mid-1; }

```
mid = (first + last) / 2;
}
if (first > last) {
    S.O.P ("Element is not found");
}
}

Public static void main (String args[]) {
int arr[] = {10, 20, 30, 40};
int key = 30;
int last = arr.length - 1;
    binarySearch (arr, 0, last, key);
}
}
```

O/P: Element is found at index : 2