Assignment - 7.

1) Explain Encapsulation with an example.

→ In Java Encapsulation is the process wrapping or hidding data in single unit for protecting the data from unnecessary user.

we can create fully encapsulated class by making the data private and now used the getter and setter method to set and get the data in it.

eg :-> // simple example with only one field

```
Package assignment7;
public Class Student {

private String name;

public String getName(){
  return name;
}

public Void SetName(String name){

this.name = name
}
}
```

```
Package assignment 7;
class Test {
public static void main (String[] args) {
Student s = new Student();
S. SetName ("gayatri");
System.output.println (s.getName());
}
}
```

Output :- gayatri.

2) Explain naming Convention for Java Beans for getter Setter method for Boolean and non Boolean types

→ In Java Bean the naming Convension for getter method are the names are Com-posed of get and for Setter method name Composed of Set, plus property name having first character is Capital.

An for boolean and not boolean instead of Get used 'is' a for getter method and Setter is Same as earlier.

3) Difference between

Default Constructor	parameterize Const
1) Constructor that is automatically produced by the Compiler in the absence of any programmer defined ~~Compiler~~. Constructor.	1) Constructor that is generated by the programmer with one or more parameter to intialize the instance Variable of class.
2) Has no parameter	2) Has one or more parameter.
3) When programmer not define any Constructor default Constructor automatically called.	3) programmer Should written own Constructor when writing a parameterized Constructor.

PAGE
DATE

4) Why == should be used for Compare objects? How else should we check for equality?

→ using == operator to Compare two object does not check to see if they have the same Value, Rather it checks to see if ~~ab~~ both object references point to exactly same object in the memory.

Although it doesn't bother us bcd ~~until~~ as long as the result we are expecting doesn't change.

So else we used Equals() method. So that you can Compare the state of two Object or the Content of the Object.

for eg :-> Java.lang.String class override the equals() and hashset Code() and in the overridden method it will check that two String Contain Same value or character if yes then they equal otherwise not equal.

5) What is the output -

```
String s = "Abc";
String S1 = new String ("abc");
S.O.P (S==S1);
S.O.P (S.equals(S1));
S.O.P (S.equalsignore case(S1));
```

→ false
false
True.

because S1 and S2 are different in the case of character so equal() method return false
but when we used equalignorecase then the case will be ignore and it gives true as a output.

6) Important of equal() and hashcode() method
→ equal() and hashcode() are the two important method in Java provide by object class for Comparing objects

1) equal() :- • it is used to Compare two objects
• To Compare two object wheather they are the same, it Compare the Value of the both object.
• By default two object same only when it store in same memory location

Syntax :- Public boolean equals (object obj)

2) hashcode() :- • hascode is a integer value a-
ssociated with every object in Java, Facilitating
the hashing in hash table.

- To get this value we used hashcode()
- hashcode() return the integer hashcode value of given class
- The hashcode() return the same hash value when called on two objects which are equal according to equal(). and if object are une-qual it usually return differeent hash values.

Syntax :- Public int hashCode()

7) different between

Comparable	Comparator
1) Comparable provide Single Sorting Seq.	1) Comparator provide multiple Sorting Seq.
2) affects the original class	2) dosen't affect original class
3) CompareTo() for sorting	3) Compare() for sorting
4) we can sort list element by Collection. Sort (List)	4) we can sort list element by Collection. Sort (List, Comparator)

35
28
1—
53

A
B
C
D
B