* static in java!.

- static is a keyword in java.
- used to define fields and methods.
- It belongs to the class not any instance.

Diff →

static	Non-static.
⇒ static is associated with class.	⇒ Associated with instance of class.
⇒ Access static fields & methods using classname	⇒ Access using reference of instance.
⇒ Get memory inside java method area.	⇒ Gets memory inside the heap section.
⇒ All static members instantiate at class loading.	⇒ Non-static members get place whenever instance is created.

Q. 1 What is the role of static in context of memory.

→ - static keyword allocates the memory for variable only once in program lifetime.
- static variable retains its value between methods call within program.

Q.2 Can static methods be overloaded and overridden in Java? How static variables shared accross multiple instances of a class?

⇒ Static Method overloading → Yes we can overload the static methods with same name & diff. parameters.

Static Method overriding → No, we cannot override the static methods because they get call using classname. we can define static method in subclass & superclass with same signature.

e.g.

```
class A{
    static Method Display();
}
class B extends A{
    static Method Displa();
}
P.S.V.M(—){
    A a = New B();
    A.display ——→ calls superclass method
    B.display ——→ calls baseless method.
```

Static variables: static variables shares accross the all instances.
- There is only one copy of a static variable in memory regardless of how many objects of the class are created.
- If any changes made to static variables it reflects in all instances.

Q.3 What is significance of final in java?

- In java final keyword is used make any constru
- once value is assigned we cannot change it.
- It indicates that we cannot modify or extends the final members.
- we can assign value at declaration or at initialization.

Page No.
Date

Assignment 4

Q. 4 What are the narrowing & widning conversions in Java

→

Narrowing: It is the process of converting the value of variable from wider datatype to narrower data type.

- In Narrowing explicit typecasting is Required.

e.g.

```
Double d = 100.10sd ;
int i = (int) d ;   ← explicit typecast.
```

```
// op → 100
```

- Here datalose is there.

Widning: It is the process of converting narrower data type to wider data type.

- No explicit type cast required.
- No data loss.

e.g.

```
int a = 100 ;
double d = a ;
```

```
//op = 100.0
```

Q. 5 Give examples of narrowing & widning.

→

i) Widning →

```
int a = 100 ;     // Here double is wider than int.
double d = a
Sysout (d)
```

⇒ O/p = 100

ii) Narrowing →

e.g.
```
double d = 10.50;
int i = d;          // Here int is narrower data type.
Sysout(i);
```
⇒ o/p → 10.

Q.6. How does Java handle potential loss of precision during narrowing conversions?

→ In java while narrowing we need to do explicit typecast otherwise compiler gives error. compiler aknowledge programmer that data might be lost during conversion.

Q.7 Explain the concept of Automatic widning conversion in java?

→ In java when we converting Narrower data type to wider data type then there is no need to explicit typecast. Compiles ~~act~~ automatically convert it. because there is no data loss.

Q.8. What are the implications of narrowing & widning conversions on type compatibility & data loss?

→

- In narrowing there might be data loss because we are converting value of larger datatype into value of smaller datatype. this is because of rounding off. So narrowing conversion compromise type compatability without data loss.

- while in widning there is no data loss., so it is totally safe as it preserves the original value.
So, widning conversion maintains the type compatibility