# Assignment - 4

Q.1) What is role of Static Keyword in Context of memory management?

→ 1) Static Keyword in Java means that the member belongs to the class, not instances of class.

2) Static Method : can be called without creating an instance of the class. It cannot acess instance variables.

3) Non-static methods : Requires an instance of a class to be called and can access both static & Non-static.

Q.2) What is role of Static Keyword in context of memory management?

→ Static method are stored in ~~to~~ method Area and shared among all instances of class. This helps in saving memory when multiple instan of a class use same data.

Q.3) Can static method be overloaded and overriden in Java? How are static variables shared across multiple instances of a class?

→ Yes, static methods can be overloaded. It can have multiple methods with same name but diff parameters.

Static methods cannot be overridden, can be hidden by Subclass with same name.

Static variables are shared all instances of class.

All instances access same static variables, which is why it's useful for storing class-wide data.

Q.4) What is Significance of final keyword in Java?

→ Final variable : once initialized, cannot be changed

Final Method : cannot be overriden by subclasses, ensuring that the method's behaviour's remains consistent.

Final class : cannot be subclassed, meaning no other class can extend it.

Q.5) What are narrowing and widening Conversion?

→ Widening Conversion : Auto conversion from smaller type to a larger type. Java handles this implicitly and saftely.

Narrowing Conversion : Explicit Conversion from a larger type to a smaller type (eg double to int). may cause data loss.

Q.6) Example of above :-

→
```
int a = 5;
Float b = a;  // widening
```

```
Float a = 5.67f ;
int b = (int) a ; // narrowing
```

Q.7) How does Java handle Potential loss of Precision during narrowing Conversion?

→ Java truncates value during narrowing Conversions, meaning the fractional part is discarded.

eg :- Casting 5.67 to int results in 5.

Q.8) Explain automatic widening Conversion in Java?

→ Java Automatically Converts a smaller data type to a larger data type (eg. int to double) to prevent data loss and ensure compactibility without explicit casting.

Q.9) What are implications of narrowing and widening Conversions on type compactbility and data loss?

→ Widening: Generally Safe and automatic. It does not result in data loss and is handled by Java without needing explicit casting.

Narrowing: Can cause data loss and requires explicit casting. Precision may be lost, and Some values may be accurately represented.