Mon

Camlin Page 25
Date 9 / 9 / 24

ASSIGNMENT NO : 4

Q.1 What does the static keyword mean in Java? Explain the difference between static & non-static methods.

→ Static methods can be called without creating an object of class and can only access static variables.

Non-static methods require an object to be called and can access both static & instance variables.

Q.2 What is the role of static keyword in the context of memory management?

→ In memory management, the static keyword ensures that a variable or method is allocated once and shared among all instances of a class. Static variables are stored in static memory area for lifetime of program.

Q.3 What is the significance of final keyword in Java?

→ In Java, the final keyword signifies that a variable's value cannot be changed once assigned, a method cannot be overridden by subclass, and a class cannot be subclassed.

Q.4 Can static methods be overloaded and overridden in Java? How static variables shared across multiple instances of a class?

→ Static methods in Java can be overloaded but not overridden. They are shared across multiple instances of a class through common memory location.

Q.5 What are narrowing and widening conversions in Java?

→ Widening conversion refers to converting smaller data type to larger one [int to long] & is done automatically. Narrowing conversion refers to converting larger data type to smaller one [long to int] & requires type casting.

Q.6 Provide examples of narrowing and widening conversions between primitive data types.

→
```
int num = 10;
long l = num;          } widening

double d = 122.7;
int i = (int) d;       } narrowing
```

Q.7 How does Java handle potential loss of precision during narrowing conversions?

→ It handles it by explicit casting from a larger to smaller data type. For given example

```
double d = 222.27;
int val = (int) d;
```

it loses fractional part.

Q.8 Explain the concept of automatic widening conversion in Java.

→ When larger data type has value of smaller data type or when smaller data type is converted into larger data type, this conversion is handled by Java compiler implicitly. It ensures: i) No data loss
ii) Automatic Handling.

Q 9 What are implications of narrowing & widening conversions on type compatability and data loss?

→ ❖ Implications of Narrowing:

(A) Type Compatability
(B) Data Loss

❖ Implications of Widening:

(A) Type Compatability
(B) No Data Loss
