CDAC Mumbai PG-DAC August 24

Assignment No- 4

1) Write a program that demonstrates widening conversion from int to double and prints the result.

Ans:

```
public class Qs1 {
   public static void main(String[] args) {
      int intValue = 100; // Integer value
      double doubleValue = intValue; // Widening conversion from int
to double
      System.out.println("Integer value: " + intValue);
      System.out.println("Double value: " + doubleValue);
   }
}
```

Qs1.java > ...

```
1   public class Qs1 {
        Run | Debug
2       public static void main(String[] args) {
3           int intValue = 100; // Integer value
4           double doubleValue = intValue; // Widening conversion from int to double
5
6           System.out.println("Integer value: " + intValue);
7           System.out.println("Double value: " + doubleValue);
8       }
9   }
10
```

CDAC MUMBAI

```
PS C:\Users\Sumit\Downloads\Assignment 4> javac Qs1.java
PS C:\Users\Sumit\Downloads\Assignment 4> java Qs1
Integer value: 100
Double value: 100.0
PS C:\Users\Sumit\Downloads\Assignment 4>
```

2) Create a program that demonstrates narrowing conversion from double to int and prints the result.

Ans:

```
public class Qs2 {
   public static void main(String[] args) {
      // Declare a double variable
      double doubleValue = 123.45;
     // Perform narrowing conversion from double to int
      int intValue = (int) doubleValue;
     // Print the original double value and the converted int value
      System.out.println("Original double value: " + doubleValue);
      System.out.println("Converted int value: " + intValue);
   }
}
```

CDAC MUMBAI

```
Qs2.java > ...
public class Qs2 {
    Run | Debug
    public static void main(String[] args) {
        // Declare a double variable
        double doubleValue = 123.45;

        // Perform narrowing conversion from double to int
        int intValue = (int) doubleValue;

        // Print the original double value and the converted int value
        System.out.println("Original double value: " + doubleValue);
        System.out.println("Converted int value: " + intValue);
    }
}
```

```
PS C:\Users\Sumit\Downloads\Assignment 4> javac Qs2.java
PS C:\Users\Sumit\Downloads\Assignment 4> java Qs2
Original double value: 123.45
Converted int value: 123
PS C:\Users\Sumit\Downloads\Assignment 4>
```

3) Write a program that performs arithmetic operations involving different data types (int, double, float) and observes how Java handles widening conversions automatically.

Ans:

```
public class Qs3 {
    public static void main(String[] args) {
        // Declare variables of different data types
```

```
    int intValue = 10;
    double doubleValue = 5.5;
    float floatValue = 3.3f;
    // Perform arithmetic operations
    double result1 = intValue + doubleValue; // int to double
    float result2 = intValue + floatValue;   // int to float
    double result3 = floatValue + doubleValue; // float to double
   // Print the results
    System.out.println("Result of int + double: " + result1);
    System.out.println("Result of int + float: " + result2);
    System.out.println("Result of float + double: " + result3);
  }
}
```

Qs3.java > ...

```java
public class Qs3 {
    Run | Debug
    public static void main(String[] args) {
        // Declare variables of different data types
        int intValue = 10;
        double doubleValue = 5.5;
        float floatValue = 3.3f;

        // Perform arithmetic operations
        double result1 = intValue + doubleValue; // int to double
        float result2 = intValue + floatValue;   // int to float
        double result3 = floatValue + doubleValue; // float to double

        // Print the results
        System.out.println("Result of int + double: " + result1);
        System.out.println("Result of int + float: " + result2);
        System.out.println("Result of float + double: " + result3);
    }
}
```

```
PS C:\Users\Sumit\Downloads\Assignment 4> javac Qs3.java
PS C:\Users\Sumit\Downloads\Assignment 4> java Qs3
Result of int + double: 15.5
Result of int + float: 13.3
Result of float + double: 8.799999952316284
PS C:\Users\Sumit\Downloads\Assignment 4>
```

4) Write a Program that demonstrates widening conversion from int to (double,float, boolean, string) and prints the result.

Ans:

```
public class Qs4 {
   public static void main(String[] args) {
      // Declare an int variable
      int intValue = 42;
      // Widening conversion from int to double
      double doubleValue = intValue;
      // Widening conversion from int to float
      float floatValue = intValue;
      // Conversion from int to String
      String stringValue = Integer.toString(intValue);
      // Manual conversion from int to boolean (example: non-zero is
true, zero is false)
      boolean booleanValue = (intValue != 0);
      // Print the results
      System.out.println("Original int value: " + intValue);
      System.out.println("Converted to double: " + doubleValue);
      System.out.println("Converted to float: " + floatValue);
      System.out.println("Converted to String: " + stringValue);
      System.out.println("Converted to boolean: " + booleanValue);
   }
```

CDAC MUMBAI

}

Qs4.java > ...

```java
public class Qs4 {
    Run | Debug
    public static void main(String[] args) {
        // Declare an int variable
        int intValue = 42;

        // Widening conversion from int to double
        double doubleValue = intValue;

        // Widening conversion from int to float
        float floatValue = intValue;

        // Conversion from int to String
        String stringValue = Integer.toString(intValue);

        // Manual conversion from int to boolean (example: non-zero is true, zero is false)
        boolean booleanValue = (intValue != 0);

        // Print the results
        System.out.println("Original int value: " + intValue);
        System.out.println("Converted to double: " + doubleValue);
        System.out.println("Converted to float: " + floatValue);
        System.out.println("Converted to String: " + stringValue);
        System.out.println("Converted to boolean: " + booleanValue);
    }
}
```

```
PS C:\Users\Sumit\Downloads\Assignment 4> javac Qs4.java
PS C:\Users\Sumit\Downloads\Assignment 4> java Qs4
Original int value: 42
Converted to double: 42.0
Converted to float: 42.0
Converted to String: 42
Converted to boolean: true
PS C:\Users\Sumit\Downloads\Assignment 4>
```

INTERVIEW QUESTIONS

Note: Write down this interview question on your notebook ,Take a screenshort & Paste that SS in the word document & upload on your Github.
What does the static keyword mean in Java? Explain the difference between static and non-static methods.

1. What is the role of the static keyword in the context of memory management.
2. Can static methods be overloaded and overridden in Java?Howstatic variables shared across multiple instances of a class?
3. What is the significance of the final keyword in Java?
4. What are narrowing and widening conversions in Java?
5. Provide examples of narrowing and widening conversions between primitive data types.
6. How does Java handle potential loss of precision during narrowing conversions?
7. Explain the concept of automatic widening conversion in Java.
8. What are the implications of narrowing and widening conversions on type compatibility and data loss?

CDAC MUMBAI

PAGE No.
DATE / /

Q7. Explain the concept of automatic widening conversion in Java.

→ Automatic conversion occurs when a value of a smaller data type is assigned to a larger data type without explicit casting. This is done automatically by the Java compiler because it is guaranteed that widening conversions do not lose data.

- For example,
  Assigning an int to a long is automatically handled by Java, as a long can hold all possible values of int.

Q8. What are the implications of narrowing and widening conversion on type compatibility and data loss?

→ Widening conversion - These are generally safe as they increase the capacity of the data type, making it compatible with the larger type without loss of info.

Narrowing conversion -

These can lead to data loss or truncation if the value exceeds the capacity of the target type. For example, converting a double with a large value to an int will result in truncation of the fractional and possible loss of precision. Narrowing requires explicit casting to make the programmer aware of potential data loss.

PAGE No.
DATE / /

Static variables sharing:

Static variables are shared across all instances of a class. They are essentially global to all instances of the class. Any and changes to a static variable in one instance are visible to all other instances.

Q3. What is the significance of the 'final' in Java?

→ final keyword in Java has several uses:

→ final variables: Once assigned, a final variable's value cannot be changed (i.e it becomes constant).

→ final methods: A final method cannot be overriden by subclass, ensuring that the methods implementation remains unchanged.

→ final classes: A final class cannot be subclassed. This ensures that the class cannot be extended, which can be useful for security reasons or to prevent misuse.

Q4. What are narrowing and widening conversions?

→ Widening Conversion:

- This refers to converting a smaller data type to a larger data type (eg int to long) widening conversions are safe and implicit as they do not lose info.

Narrowing Conversion:

- This refers to converting a larger data type to a smaller data type (eg double to int). Narrowing

conversions are potentially unsafe as they lose information and require explicit casting.

Q5. Provide examples of narrowing and widening conversions between primitive data types.

→

Widening Conversion

```
int intValue = 100;
long longValue = intValue;   // int to long
```

Narrowing Conversion

```
double doubleValue = 123.45;
int intValue = (int) doubleValue;   // double to int
```

Q6. How does Java handle potential loss of precision loss of precision during narrowing conversions?

→ Java handles potential loss of precision during narrowing conversions by requiring explicit casting for that we need to manually cast the value from larger to smaller type.
This will make potential data loss explicit and Java will not perform the conversion automatically to prevent unintended loss of info.

PAGE No.
DATE / /

Q7. Explain the concept of automatic widening conversion in Java.

→ Automatic conversion occurs when a value of a smaller data type is assigned to a larger data type without explicit casting. This is done automatically by the Java compiler because it is guaranteed that widening conversions do not lose data.

- For example,
  Assigning an int to a long is automatically handled by Java, as a long can hold all possible values of int.

Q8. What are the implications of narrowing and widening conversion on type compatibility and data loss?

→ Widening conversion - These are generally safe as they increase the capacity of the data type, making it compatible with the larger type without loss of info.

Narrowing conversion -

These can lead to data loss or truncation if the value exceeds the capacity of the target type. For example, converting a double with a large value to an int will result in truncation of the fractional and possible loss of precision. Narrowing requires explicit casting to make the programmer aware of potential data loss.