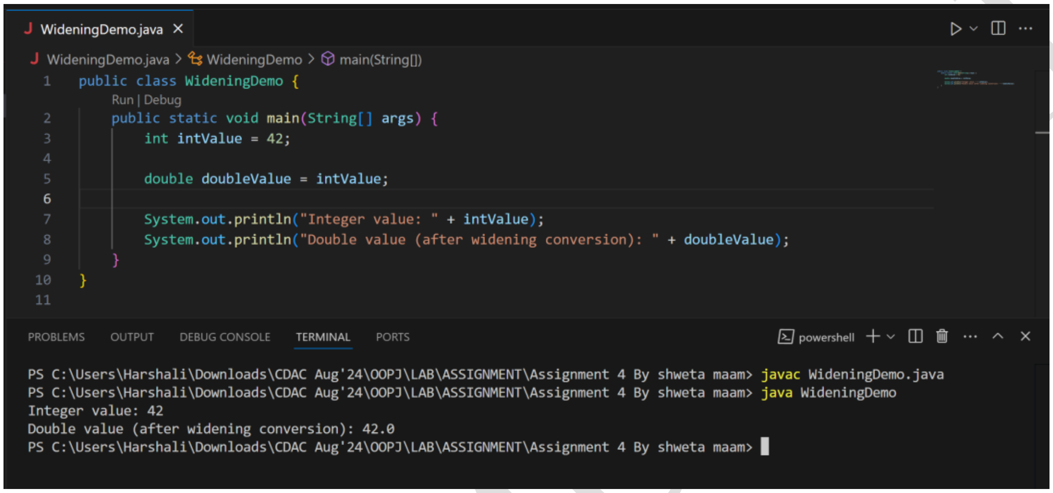
CDAC Mumbai PG-DAC August 24

Assignment No- 4

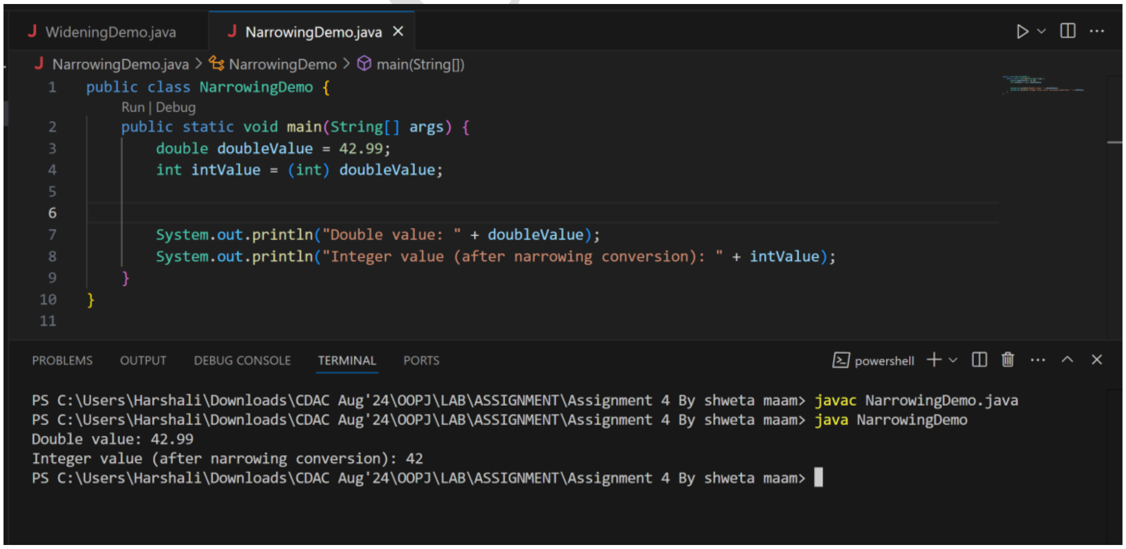
1) Write a program that demonstrates widening conversion from int to double and prints the result.


WideningDemo.java
WideningDemo.java > WideningDemo > main(String[])
Run | Debug
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell

```java
public class WideningDemo {
    public static void main(String[] args) {
        int intValue = 42;

        double doubleValue = intValue;

        System.out.println("Integer value: " + intValue);
        System.out.println("Double value (after widening conversion): " + doubleValue);
    }
}

```

```
PS C:\Users\Harshali\Downloads\CDAC Aug'24\OOPJ\LAB\ASSIGNMENT\Assignment 4 By shweta maam> javac WideningDemo.java
PS C:\Users\Harshali\Downloads\CDAC Aug'24\OOPJ\LAB\ASSIGNMENT\Assignment 4 By shweta maam> java WideningDemo
Integer value: 42
Double value (after widening conversion): 42.0
PS C:\Users\Harshali\Downloads\CDAC Aug'24\OOPJ\LAB\ASSIGNMENT\Assignment 4 By shweta maam> 
```

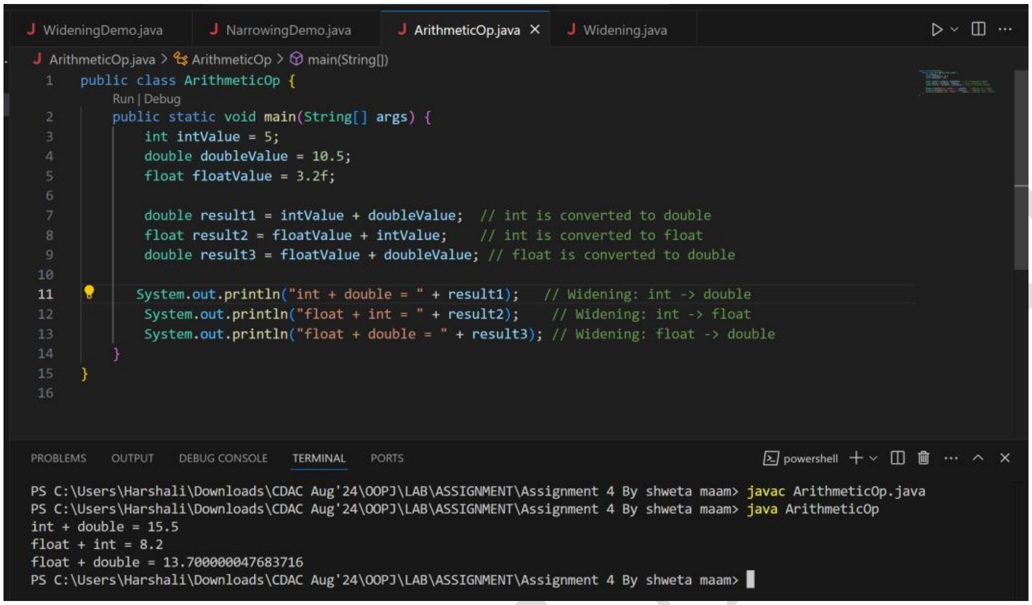
2) Create a program that demonstrates narrowing conversion from double to int and prints the result.


WideningDemo.java
NarrowingDemo.java
NarrowingDemo.java > NarrowingDemo > main(String[])
Run | Debug
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell

```java
public class NarrowingDemo {
    public static void main(String[] args) {
        double doubleValue = 42.99;
        int intValue = (int) doubleValue;


        System.out.println("Double value: " + doubleValue);
        System.out.println("Integer value (after narrowing conversion): " + intValue);
    }
}

```

```
PS C:\Users\Harshali\Downloads\CDAC Aug'24\OOPJ\LAB\ASSIGNMENT\Assignment 4 By shweta maam> javac NarrowingDemo.java
PS C:\Users\Harshali\Downloads\CDAC Aug'24\OOPJ\LAB\ASSIGNMENT\Assignment 4 By shweta maam> java NarrowingDemo
Double value: 42.99
Integer value (after narrowing conversion): 42
PS C:\Users\Harshali\Downloads\CDAC Aug'24\OOPJ\LAB\ASSIGNMENT\Assignment 4 By shweta maam> 
```

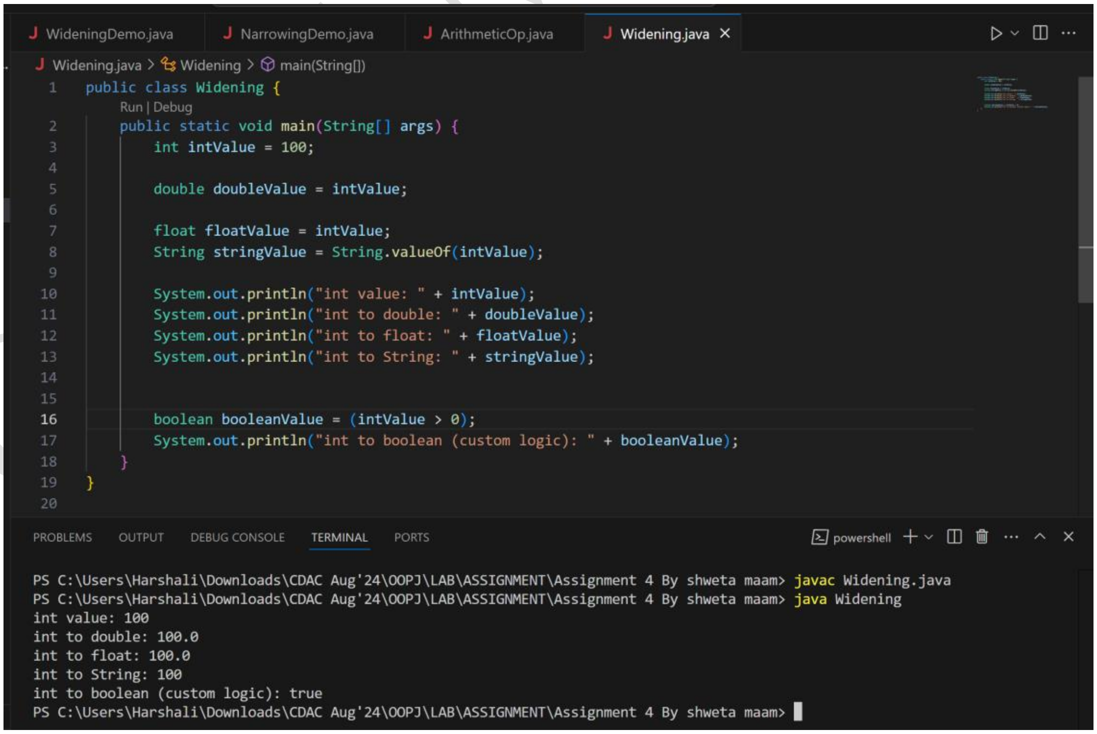
3) Write a program that performs arithmetic operations involving different data types (int, double, float) and observes how Java handles widening conversions automatically.



```java
public class ArithmeticOp {
    public static void main(String[] args) {
        int intValue = 5;
        double doubleValue = 10.5;
        float floatValue = 3.2f;

        double result1 = intValue + doubleValue;  // int is converted to double
        float result2 = floatValue + intValue;    // int is converted to float
        double result3 = floatValue + doubleValue; // float is converted to double

        System.out.println("int + double = " + result1);   // Widening: int -> double
        System.out.println("float + int = " + result2);    // Widening: int -> float
        System.out.println("float + double = " + result3); // Widening: float -> double
    }
}
```

```
PS C:\Users\Harshali\Downloads\CDAC Aug'24\OOPJ\LAB\ASSIGNMENT\Assignment 4 By shweta maam> javac ArithmeticOp.java
PS C:\Users\Harshali\Downloads\CDAC Aug'24\OOPJ\LAB\ASSIGNMENT\Assignment 4 By shweta maam> java ArithmeticOp
int + double = 15.5
float + int = 8.2
float + double = 13.700000047683716
PS C:\Users\Harshali\Downloads\CDAC Aug'24\OOPJ\LAB\ASSIGNMENT\Assignment 4 By shweta maam>
```

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



```java
public class Widening {
    public static void main(String[] args) {
        int intValue = 100;

        double doubleValue = intValue;

        float floatValue = intValue;
        String stringValue = String.valueOf(intValue);

        System.out.println("int value: " + intValue);
        System.out.println("int to double: " + doubleValue);
        System.out.println("int to float: " + floatValue);
        System.out.println("int to String: " + stringValue);


        boolean booleanValue = (intValue > 0);
        System.out.println("int to boolean (custom logic): " + booleanValue);
    }
}
```

```
PS C:\Users\Harshali\Downloads\CDAC Aug'24\OOPJ\LAB\ASSIGNMENT\Assignment 4 By shweta maam> javac Widening.java
PS C:\Users\Harshali\Downloads\CDAC Aug'24\OOPJ\LAB\ASSIGNMENT\Assignment 4 By shweta maam> java Widening
int value: 100
int to double: 100.0
int to float: 100.0
int to String: 100
int to boolean (custom logic): true
PS C:\Users\Harshali\Downloads\CDAC Aug'24\OOPJ\LAB\ASSIGNMENT\Assignment 4 By shweta maam>
```

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?

① What is the role of the static keyword in the context of memory management?

Ans- The static keyword in Java is used to define class-level variables and methods. When a variable or method is declared as static, it belongs to the class rather than any instance of the class.

- This means that a single copy of the variable or method is shared across all instances of class.
- In terms of memory management, static variables are allocated memory once, when the class is loaded, and they are deallocated when the class is unloaded. This can lead to efficient memory usage for class-wide data and methods that do not need instance-specific values.

② Can static methods be overloaded and overridden in Java? How are static variables shared across multiple instances of a class?

Ans. Overloading - Yes, static methods can be overloaded in Java. Overloading occurs when multiple static methods have the same name but different parameters with the same class.

Overriding - No, static methods cannot be overridde While we can declare a static method with the same name in a subclass, it is not considered overriding because the method is bound to the class, not instance.

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

③ what is the Significance of the 'final' in Java?

Ans. 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 overridden by Subclass, ensuring that the method's 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.

④ What are narrowing and widening conversions?

Ans. Widening Conversion -

This refers to converting a smaller data type to a larger data type (e.g. into 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 ~~sima~~ smaller data type (e.g. double to int). Narrowing conversions are potentially unsafe as they may lose information and require explicit casting.

(5) Provide examples of narrowing and widening conversions between primitive data types.

Ans. Widening Conversion :

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

Narrowing Conversion :

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

(6) How does Java handle potential loss of precision during narrowing conversions?

Ans. 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 the potential data loss explicit, and Java will not perform the conversion automatically to prevent unintended loss of info.

(7) Explain the concept of automatic widening conversion in Java.

Ans. 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 an int.

(8) what are the Implications of narrowing and widening conversions on type compatibility and data loss?

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