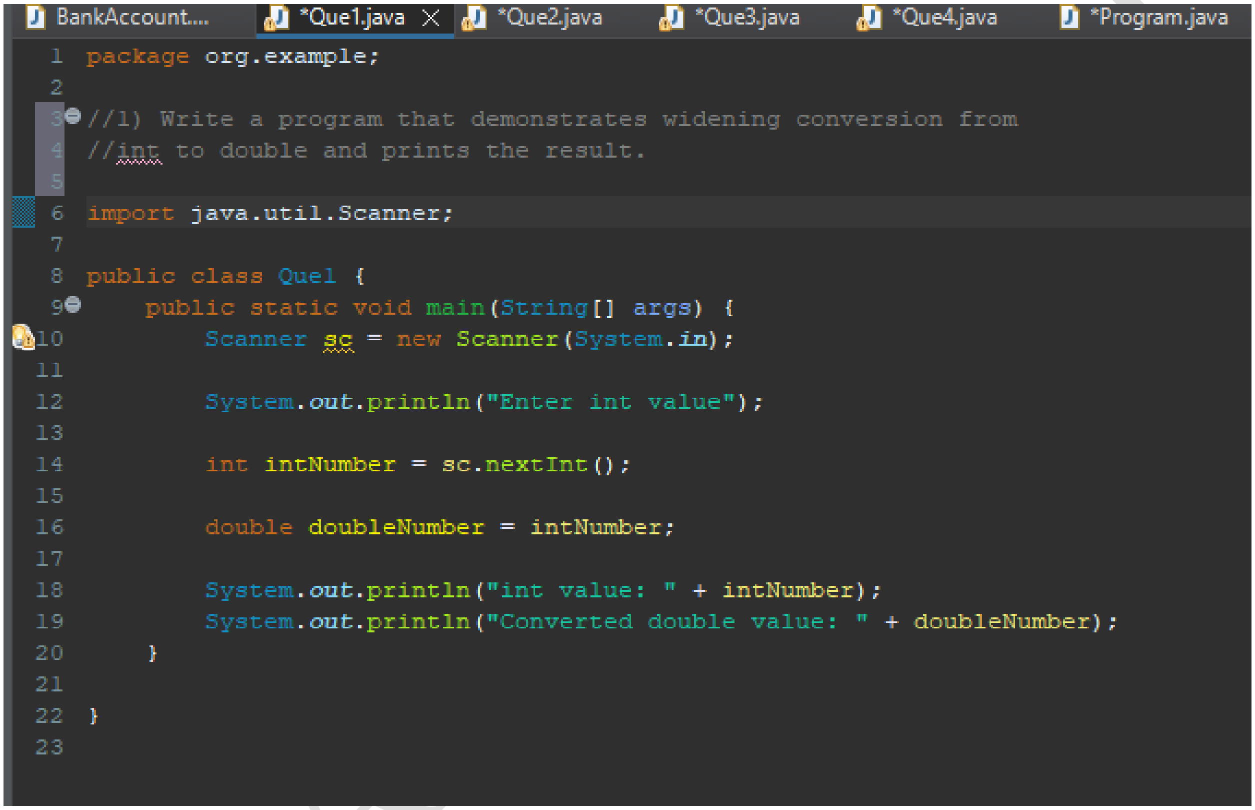
CDAC Mumbai PG-DAC August 24

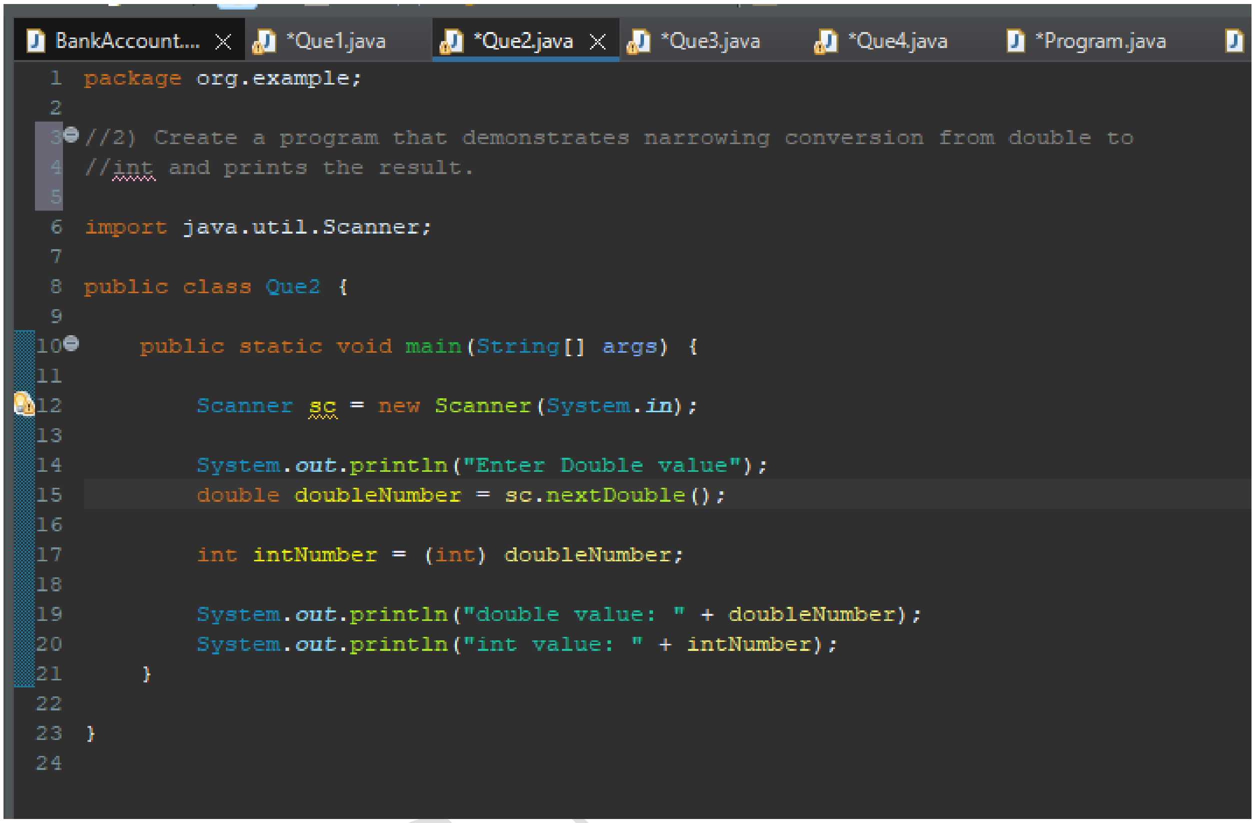
Assignment No- 4

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



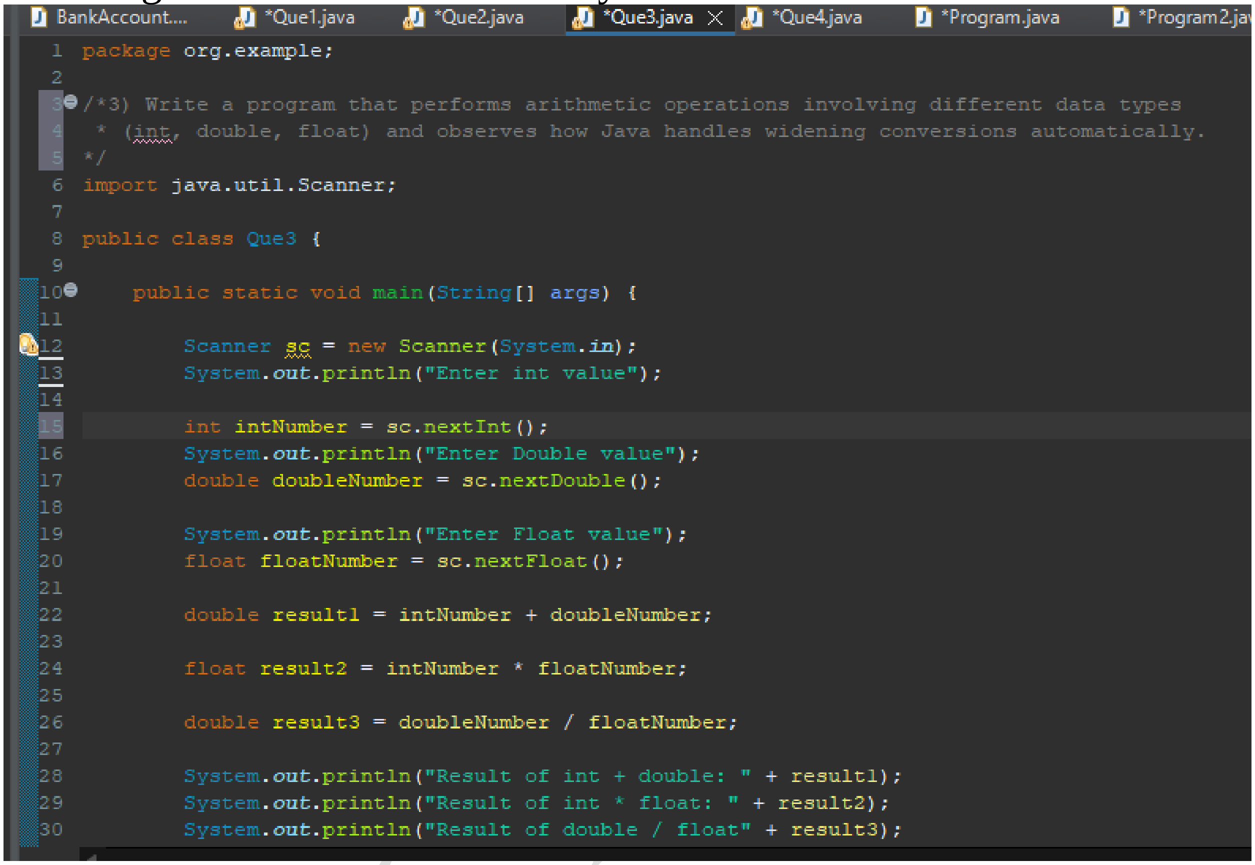
```java
package org.example;

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

import java.util.Scanner;

public class Que1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter int value");

        int intNumber = sc.nextInt();

        double doubleNumber = intNumber;

        System.out.println("int value: " + intNumber);
        System.out.println("Converted double value: " + doubleNumber);
    }

}

```

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



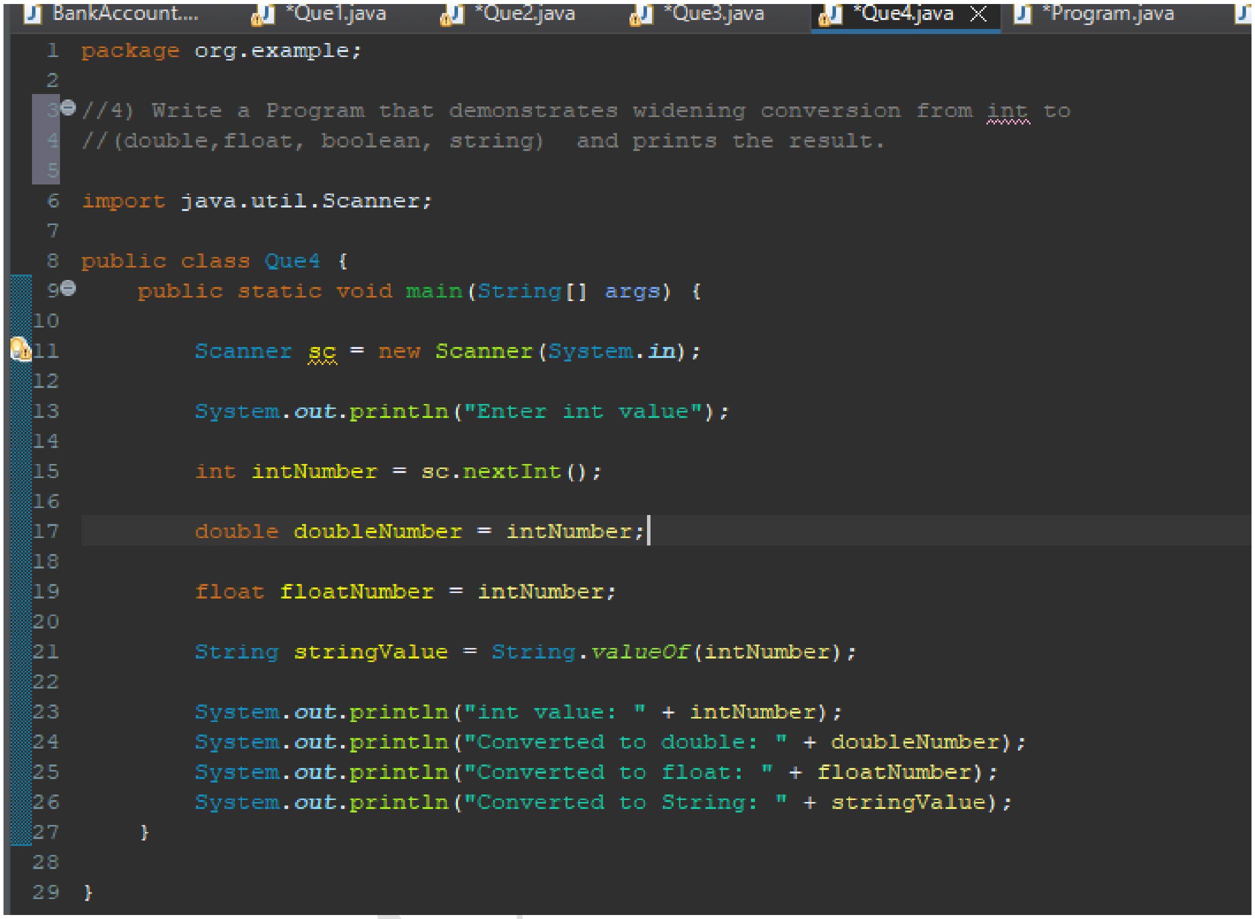
```java
package org.example;

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

import java.util.Scanner;

public class Que2 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Double value");
        double doubleNumber = sc.nextDouble();

        int intNumber = (int) doubleNumber;

        System.out.println("double value: " + doubleNumber);
        System.out.println("int value: " + intNumber);
    }

}
```

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



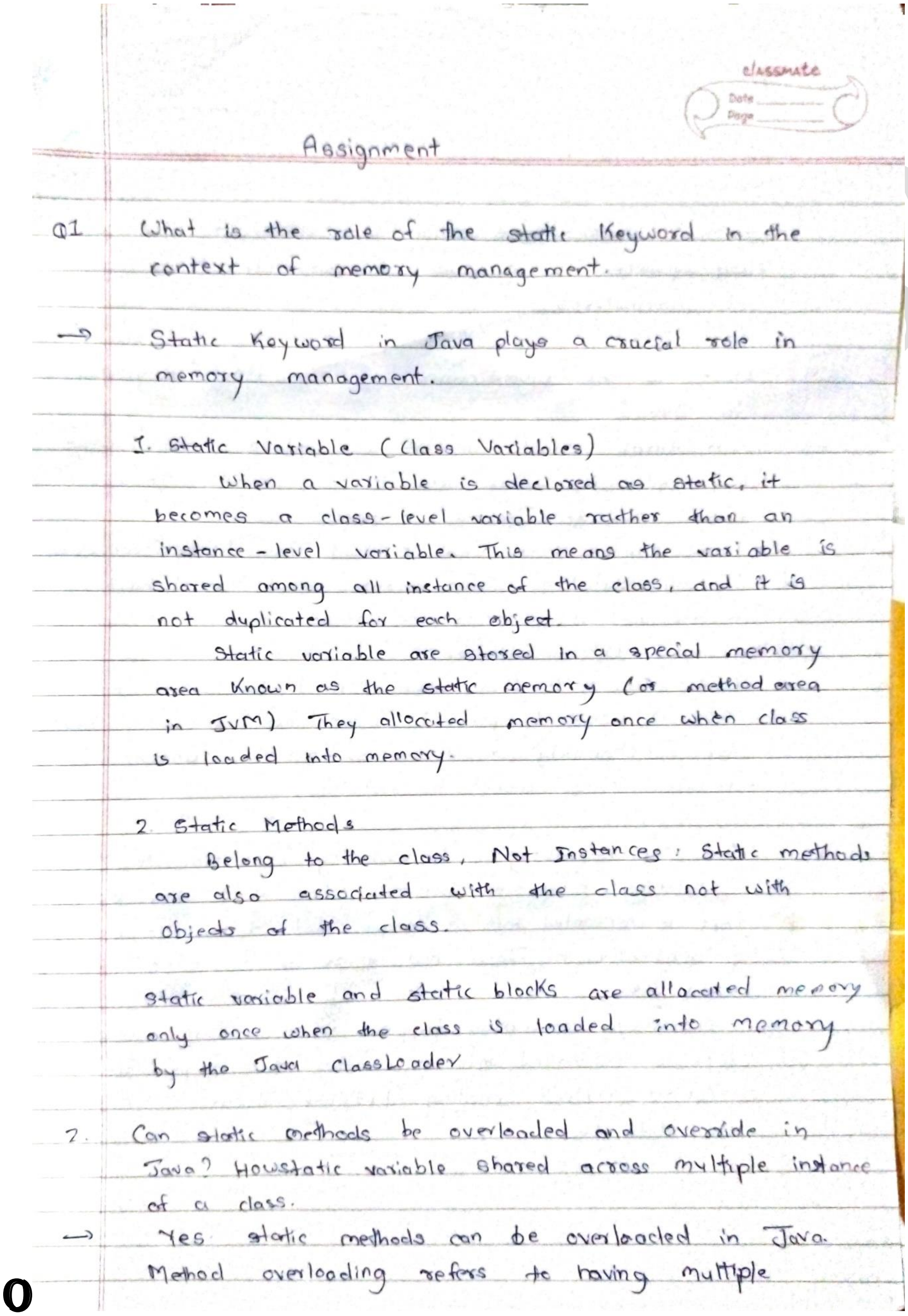
```java
package org.example;

/*3) Write a program that performs arithmetic operations involving different data types
 * (int, double, float) and observes how Java handles widening conversions automatically.
*/
import java.util.Scanner;

public class Que3 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter int value");

        int intNumber = sc.nextInt();
        System.out.println("Enter Double value");
        double doubleNumber = sc.nextDouble();

        System.out.println("Enter Float value");
        float floatNumber = sc.nextFloat();

        double result1 = intNumber + doubleNumber;

        float result2 = intNumber * floatNumber;

        double result3 = doubleNumber / floatNumber;

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

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



```java
package org.example;

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

import java.util.Scanner;

public class Que4 {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter int value");

        int intNumber = sc.nextInt();

        double doubleNumber = intNumber;

        float floatNumber = intNumber;

        String stringValue = String.valueOf(intNumber);

        System.out.println("int value: " + intNumber);
        System.out.println("Converted to double: " + doubleNumber);
        System.out.println("Converted to float: " + floatNumber);
        System.out.println("Converted to String: " + stringValue);
    }

}
```

INTERVIEW QUESTIONS



Assignment

Q1 What is the role of the static Keyword in the context of memory management.

→ Static Keyword in Java plays a crucial role in memory management.

I. Static Variable (Class Variables)

When a variable is declored as static, it becomes a class-level variable rather than an instance-level variable. This means the variable is shared among all instance of the class, and it is not duplicated for each object.

Static variable are stored in a special memory area Known as the static memory (or method area in JVM) They allocated memory once when class is loaded into memory.

2. Static Methods

Belong to the class, Not Instances: Static methods are also associated with the class not with objects of the class.

static variable and static blocks are allocated memory only once when the class is loaded into memory by the Java ClassLoader

2. Can static methods be overloaded and override in Java? Howstatic variable shared across multiple instance of a class.

→ Yes static methods can be overloaded in Java. Method overloading refers to having multiple methods with same name but different parameter lists (either by number of parameters or types of parameters).

Q3 What is the significance of the final Keyword in Java?

→ A final variable can be assigned only once. Once initialized, its value cannot be changed. It is often used to declare constants.

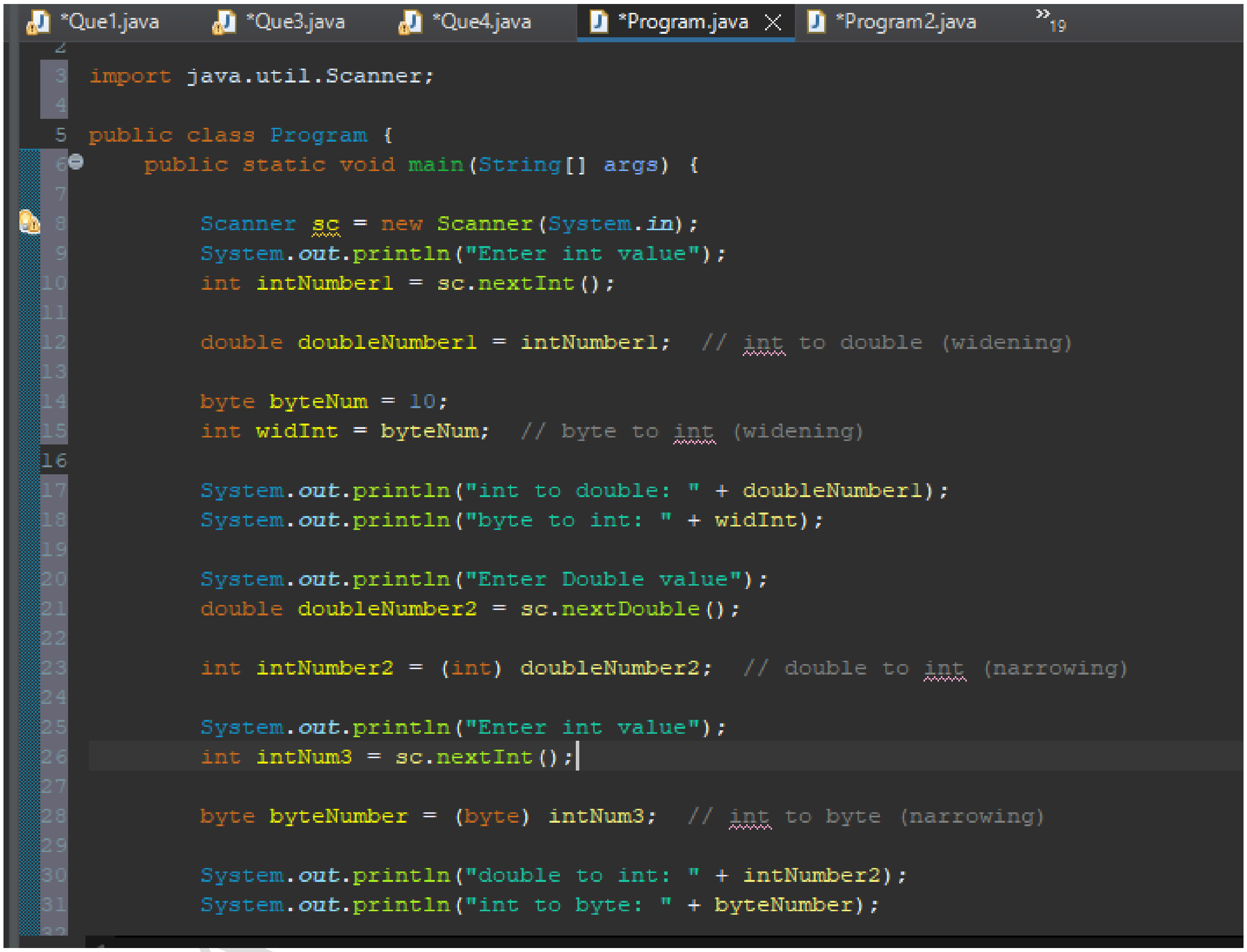
- Final Method : A final method cannot be overridden by any subclass. This ensures that the behaviour of the method remains consistent and cannot be changed by inheritance.
- Final Class : A final class cannot be subclassed. This is used to prevent inheritance, ensuring that the functionality of the class remains unchanged.
- A final class cannot be subclass.

Q4 What are narrowing and widening conversion in Java?

→ Widening conversion happens when a smaller data type is converted into a larger data type. It occurs automatically in Java and there is no risk of data loss.

Narrowing conversion happens when a larger data type is converted into a smaller data type. It requires explicit casting because there is a potential for data loss.

5 Provide examples of narrowing and2 widening conversions between primitive data types.



```
import java.util.Scanner;

public class Program {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter int value");
        int intNumber1 = sc.nextInt();

        double doubleNumber1 = intNumber1;  // int to double (widening)

        byte byteNum = 10;
        int widInt = byteNum;  // byte to int (widening)

        System.out.println("int to double: " + doubleNumber1);
        System.out.println("byte to int: " + widInt);

        System.out.println("Enter Double value");
        double doubleNumber2 = sc.nextDouble();

        int intNumber2 = (int) doubleNumber2;  // double to int (narrowing)

        System.out.println("Enter int value");
        int intNum3 = sc.nextInt();

        byte byteNumber = (byte) intNum3;  // int to byte (narrowing)

        System.out.println("double to int: " + intNumber2);
        System.out.println("int to byte: " + byteNumber);
```

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

→ Narrowing conversions involve converting a large data type (with more percision or range) to a smaller data type (with less precision or range). Because narrowing conversions can lead to loss of percision or overflow.

Java requires explicit casting to indicate that the developer is aware of the risks and intentionally allowing the conversion.

```
double doubleval = 100.11;
 int narrowval =(int) doubleval; // Explicit casting
```

Q7 Explain the concept of automatic widening conversion in Java.

→ Automatic widening conversion in Java is a feature of the language that allows a value of a smaller data type to be automatically converted to larger data type without explicit casting. This is done to preserve the value of the smaller type and ensure that no information is lost in the conversion process.

Q8 What are the implications of narrowing conversions on type compatibility and data loss?

→ Widening Conversions

Widening conversion occurs when a value of a smaller data type (eg. byte, short, int, char) is converted to a larger data type (e.g long, flat, double)

Type compatibility

Implicit Conversion

Safe conversion : Since the large data type can represent all values of the smaller type, there's no risk of data loss during conversion.

Narrowing Convesion

Narrowing conversion occurs when a value of a larger data type (eg. long, float, double) is converted to smaller data type.

Explicit Casting: Narrowing conversions require explicit casting, You must use a cast operator to tell the compiler that you are aware of the potential loss of data.