**ASSIGNMENT – 4**

**Subject: CSW2 (CSE 2141)**

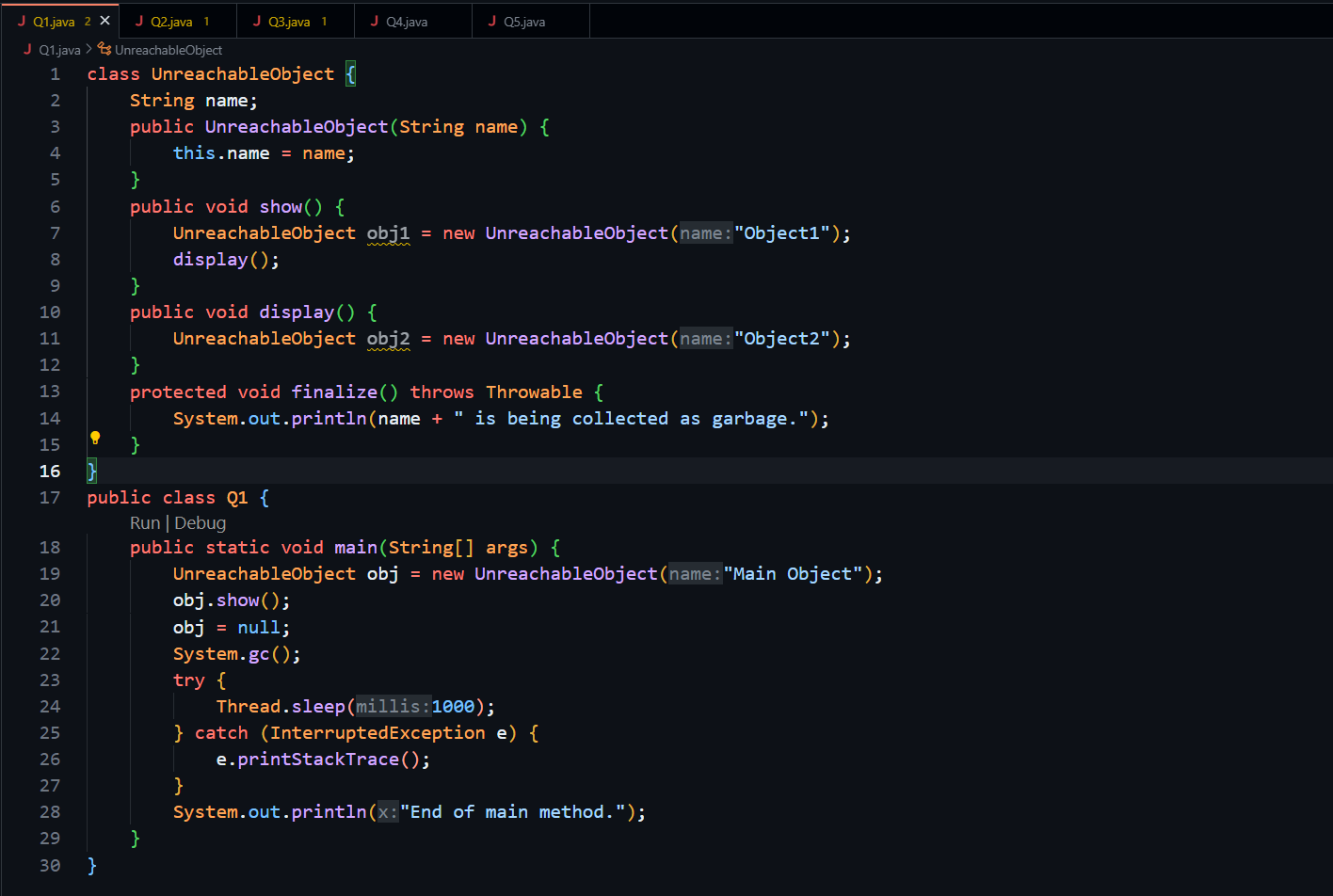
**Name: Arpit Kumar Mohanty**

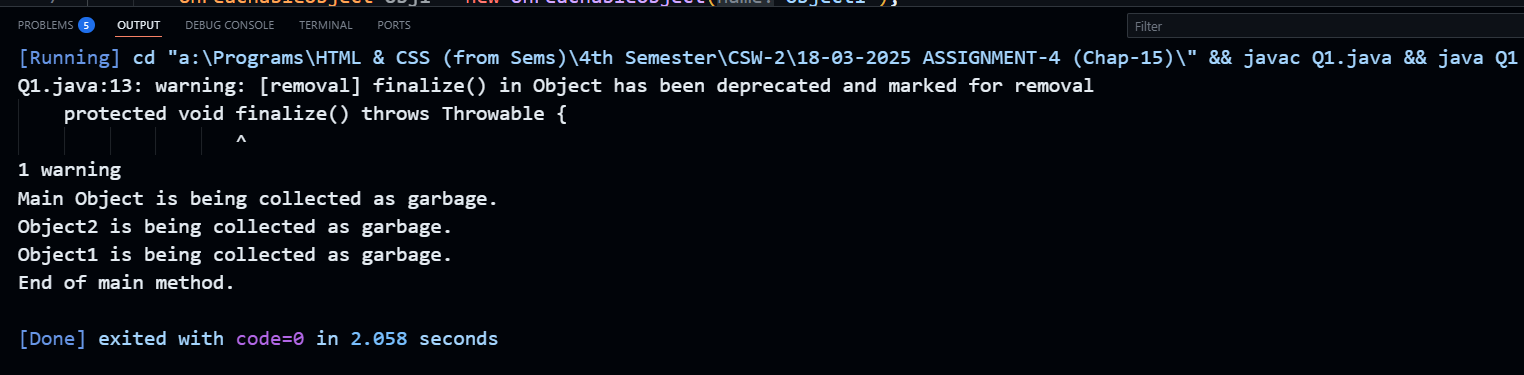
**Registration Number: 2341013237**

**Section: 23412G1**

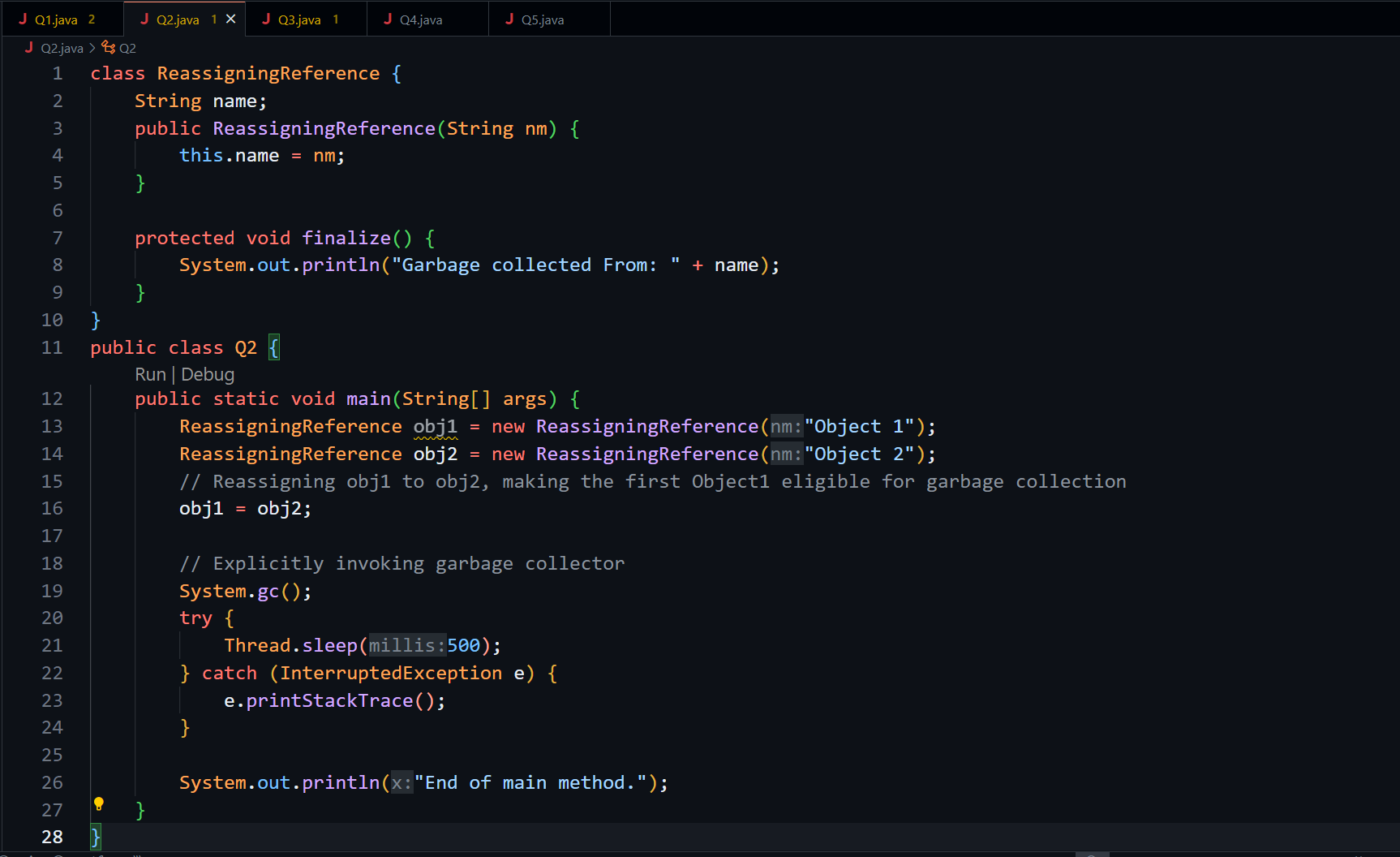
**Branch: CSE**

**Q1. Write a Java program to demonstrate garbage collection using an UnreachableObject class. It should include a constructor to initialize an object with a given name, a show() method creating an instance and invoking display(), and the display() method creating another instance. The main() method should call show() and explicitly invoke the garbage collector. The program must override the finalize() method to print the object's name upon successful garbage collection, illustrating how unreachable objects are collected.**

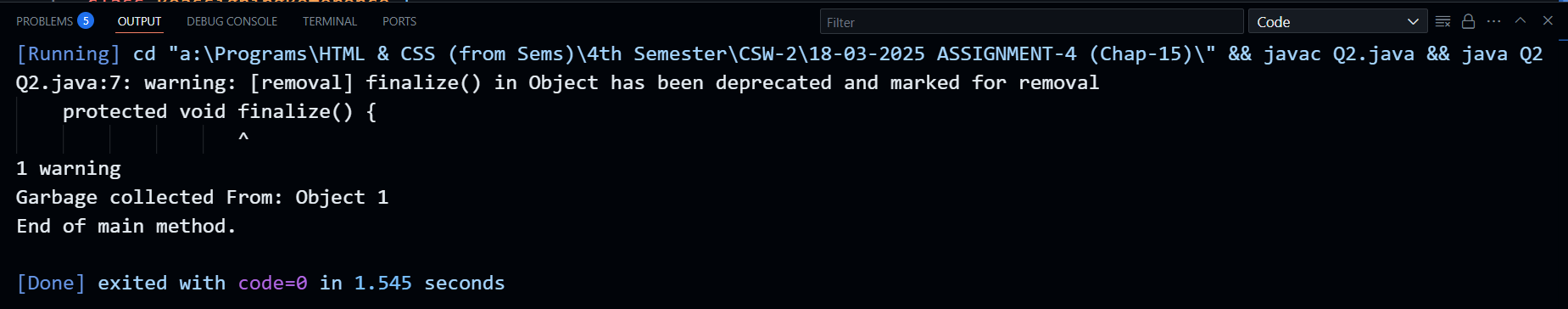
**Solution:   
Output:**

****

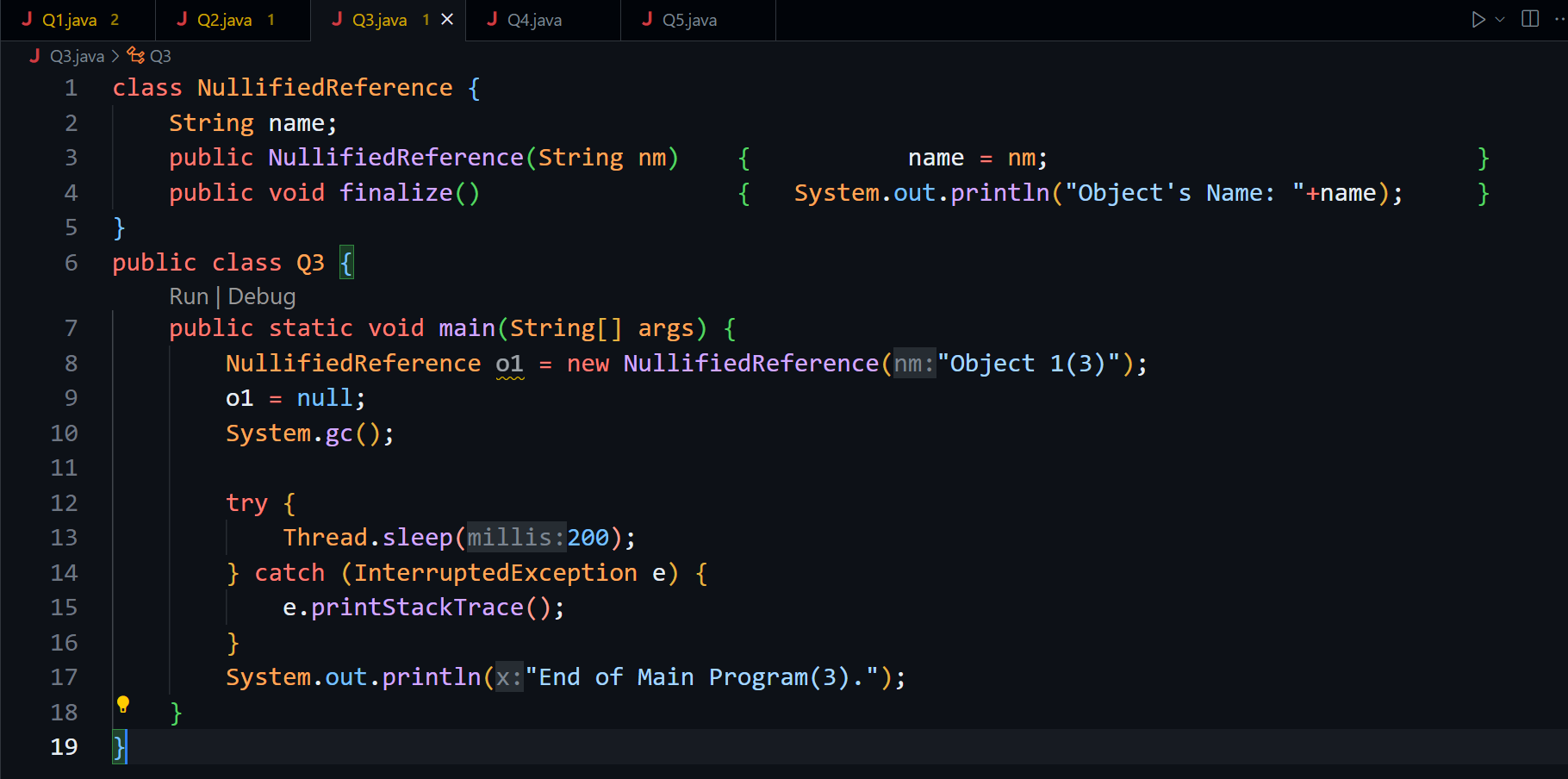
**Q2. Develop a Java program to demonstrate reference reassignment and garbage collection using the ReassigningReference class. The class should have a constructor to initialize an object with a given name. In the main() method, create two instances of ReassigningReference, reassign one reference to another, and then explicitly invoke the garbage collector. Override the finalize() method to print the object's name upon successful garbage collection.**

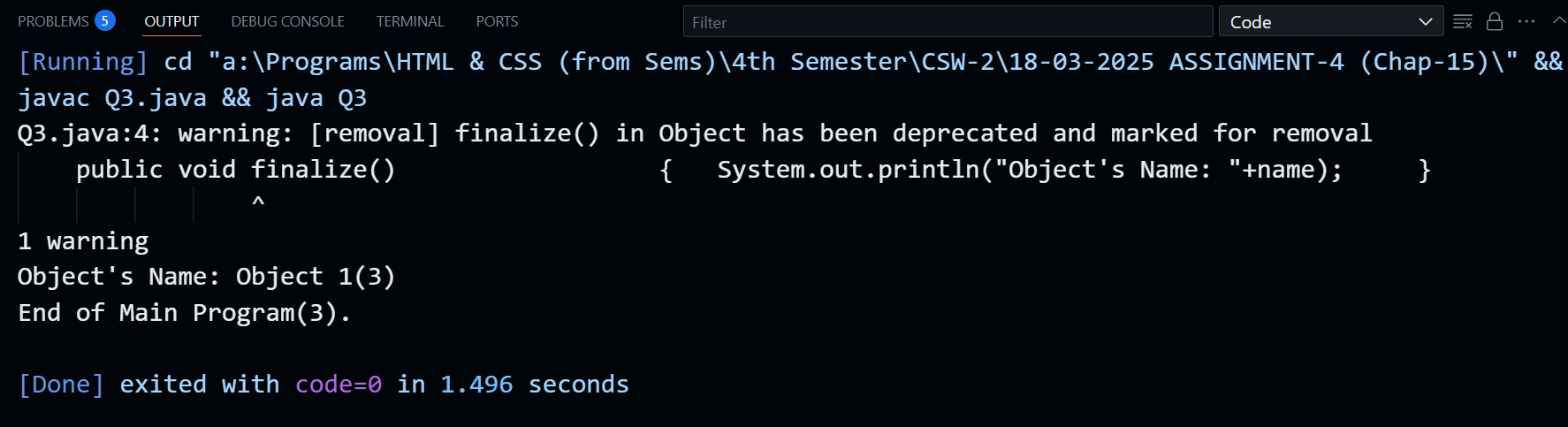
**Solution: **

**Output:**

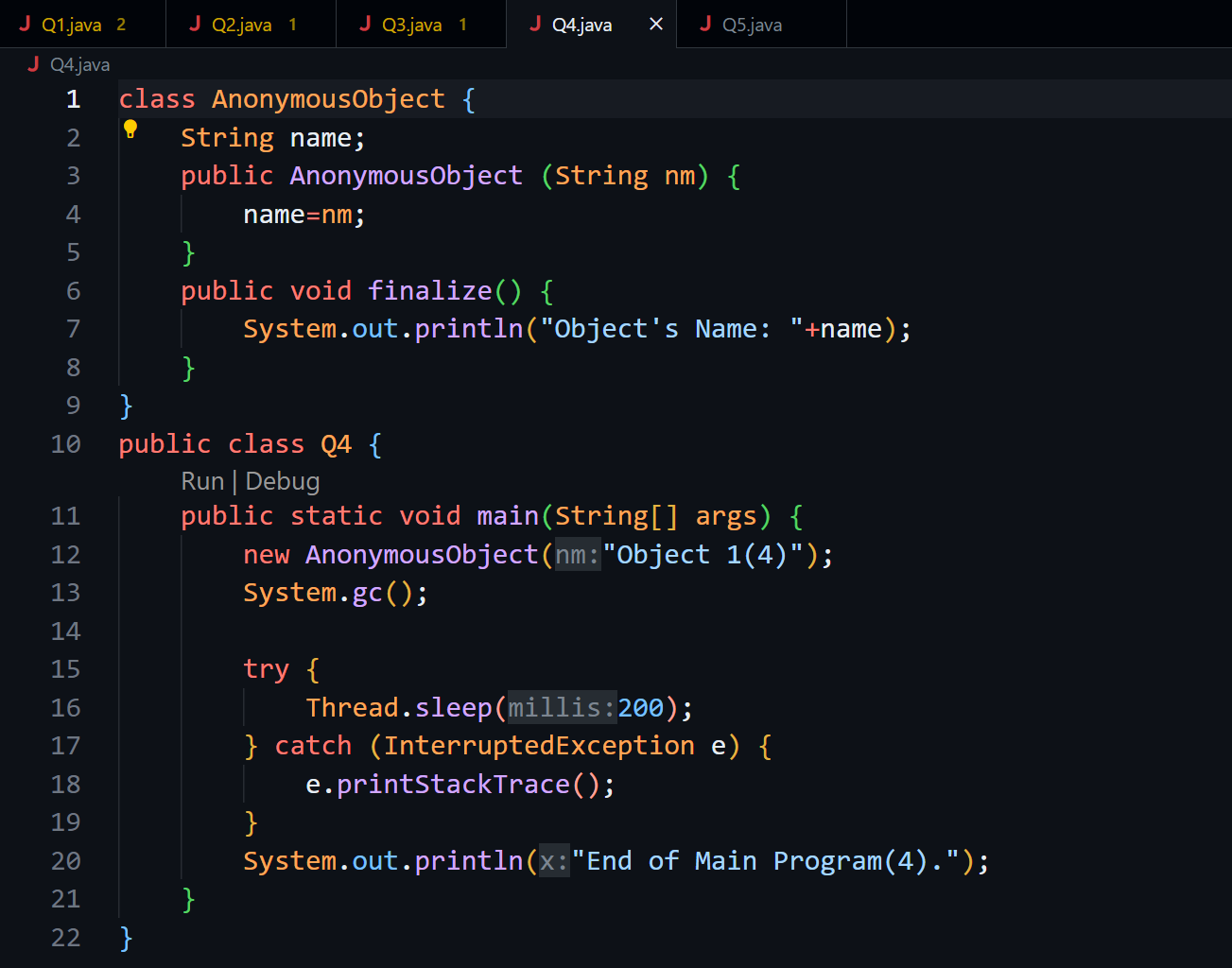
****

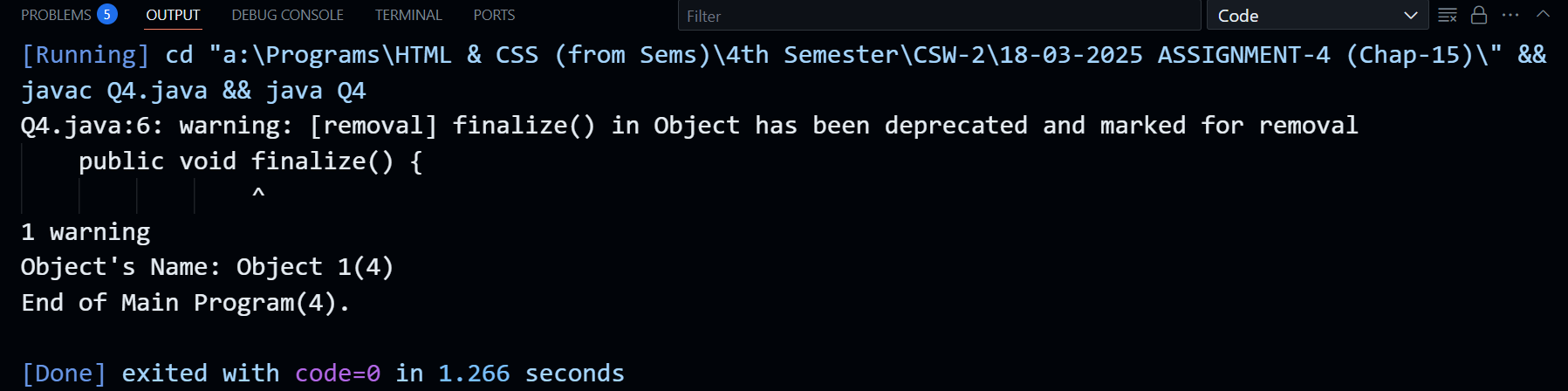
**Q3. Write a Java program to demonstrate nullification of references and garbage collection using the NullifiedReference class. The class should have a constructor to initialize an object with a given name. In the main() method, create an instance of NullifiedReference, assign it, then nullify the reference, and explicitly invoke the garbage collector. Override the finalize() method to print the object's name upon successful garbage collection.**

**Solution: **

**Output: **

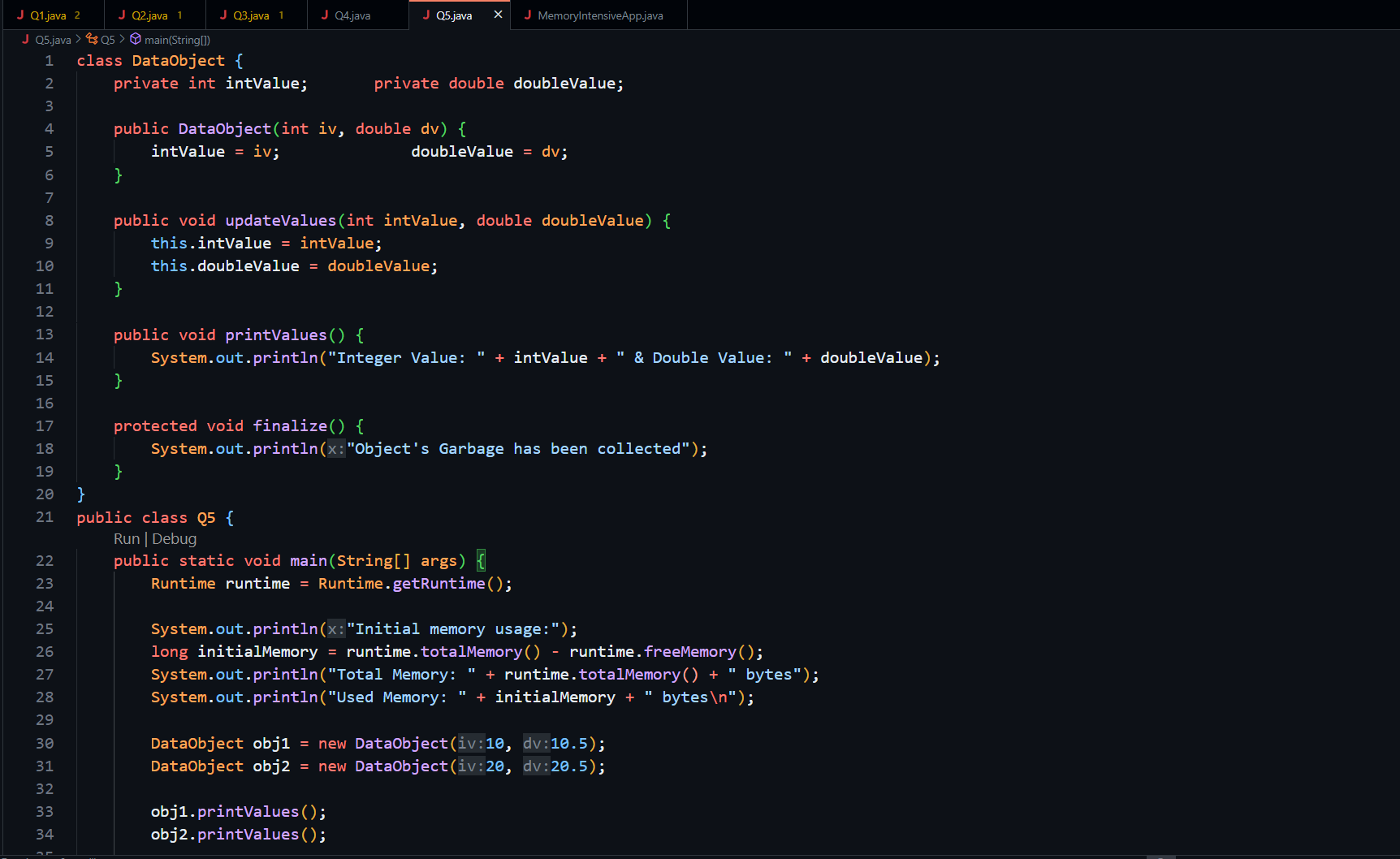
**Q4. Create a Java program to demonstrate anonymous objects and garbage collection using the AnonymousObject class. The class should have a constructor to initialize an object with a name. In the main() method, create an anonymous object and explicitly invoke the garbage collector. Override the finalize() method to print the object's name upon successful garbage collection**

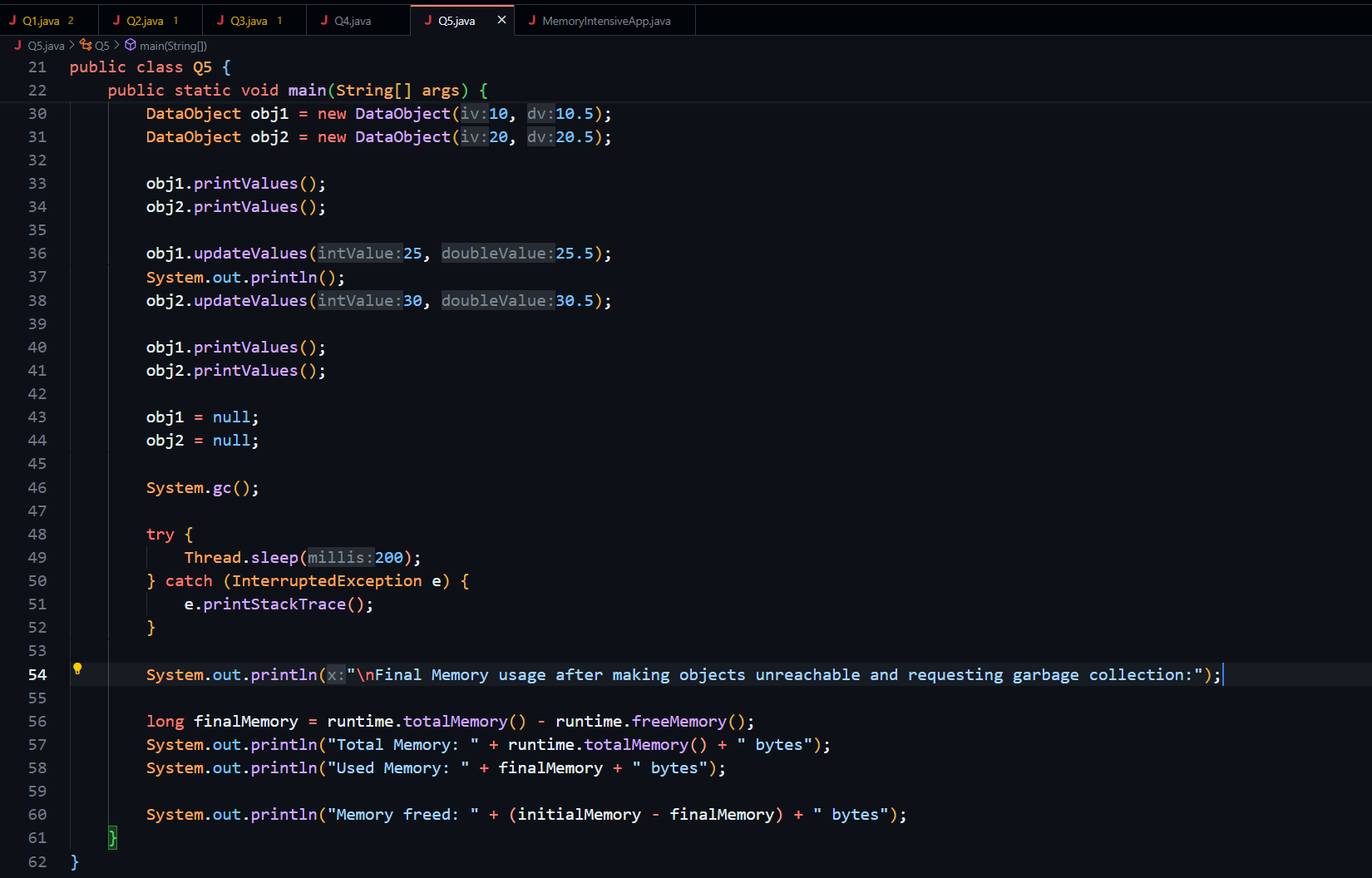
**Solution: **

**Output: **

**Q5. Develop a Java class with private integer and double data members, along with methods for initialization, setting, and updating these members. Create two objects of this class and invoke the necessary methods to modify the data. Use the Runtime class to calculate the total allocated memory and memory occupied by the objects. Apply any technique to make the objects unreachable, making them eligible for garbage collection. Finally, recheck the utilized and total memory using the Runtime class.**

**Solution:**

****

****

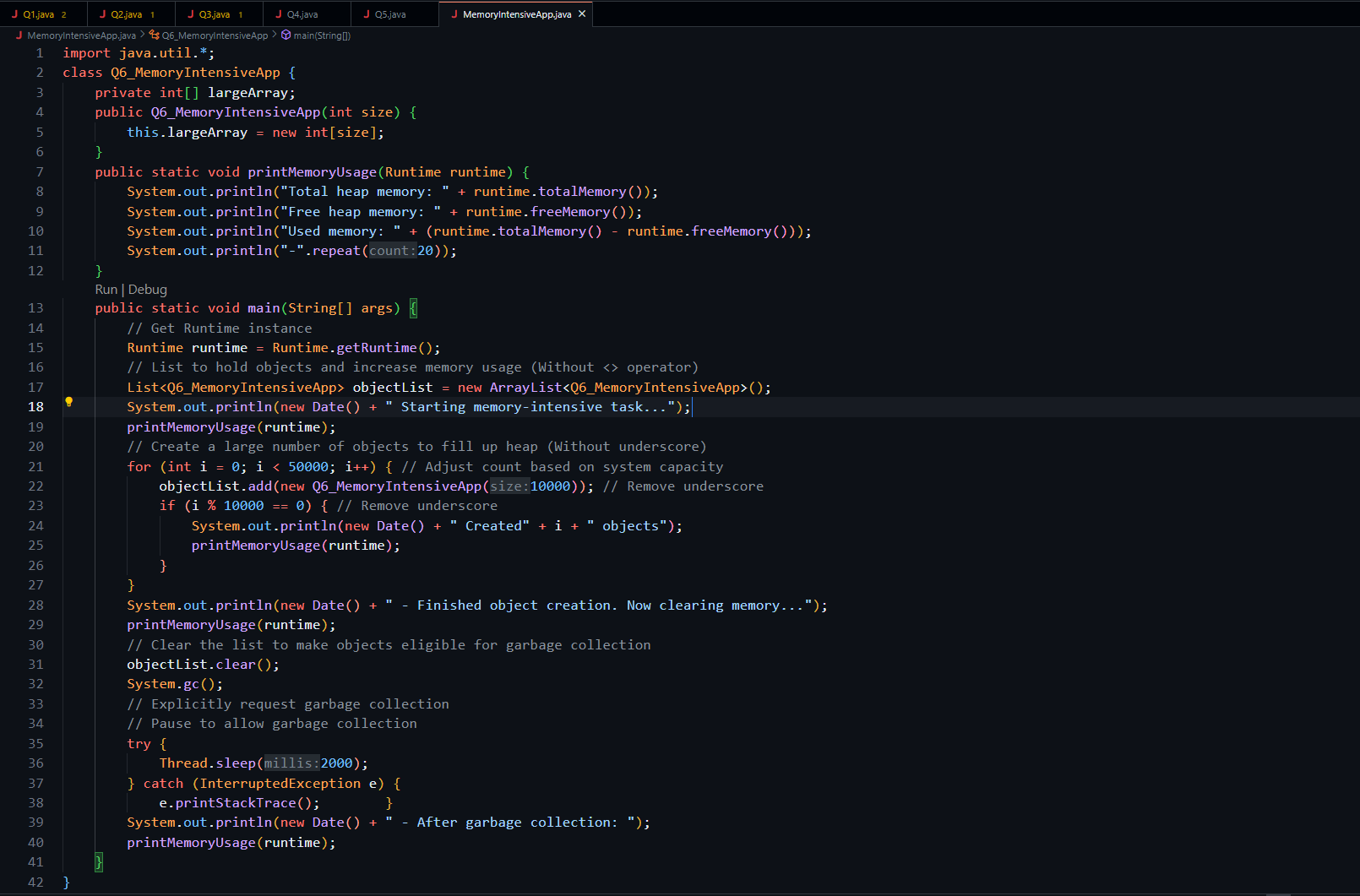
**Output:**

****

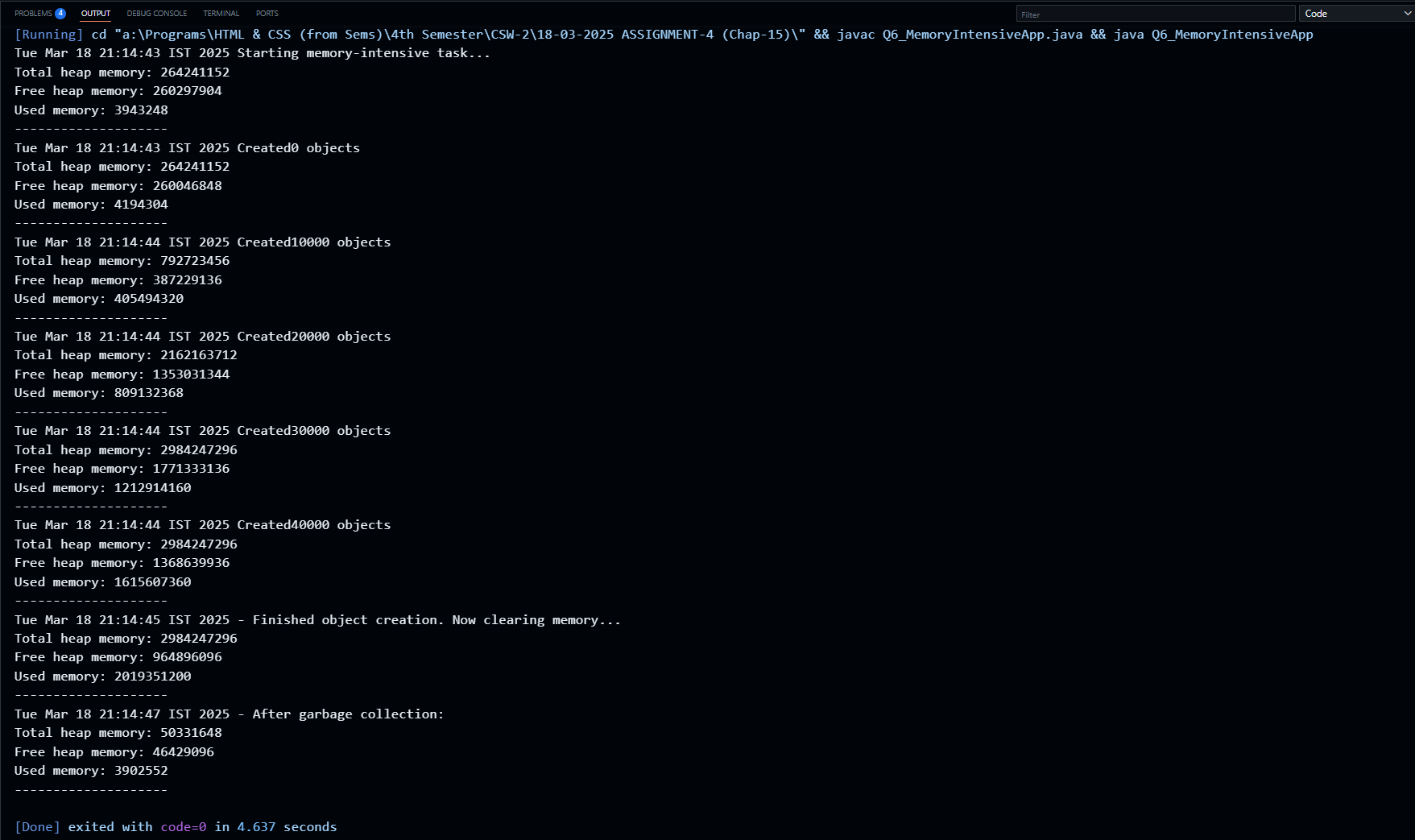
**Q6. Write a memory-intensive Java program that creates a large number of objects and test it using the G1 garbage collector. Print timestamps along with the total heap size and free memory. Use the following commands to retrieve heap memory details:**

**• Total heap memory: Runtime.getRuntime().totalMemory();**

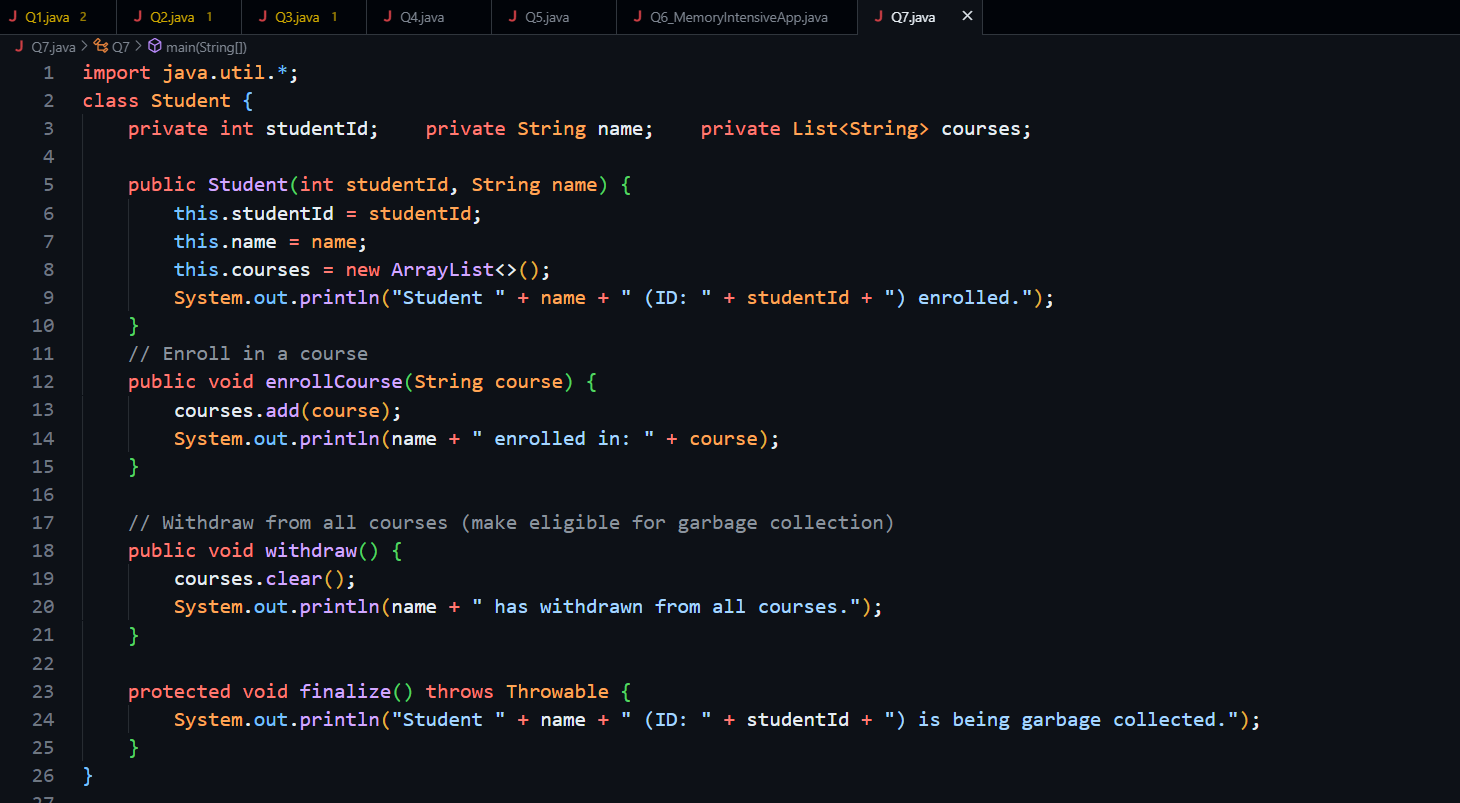
**• Free heap memory: Runtime.getRuntime().freeMemory();**

**Solution: **

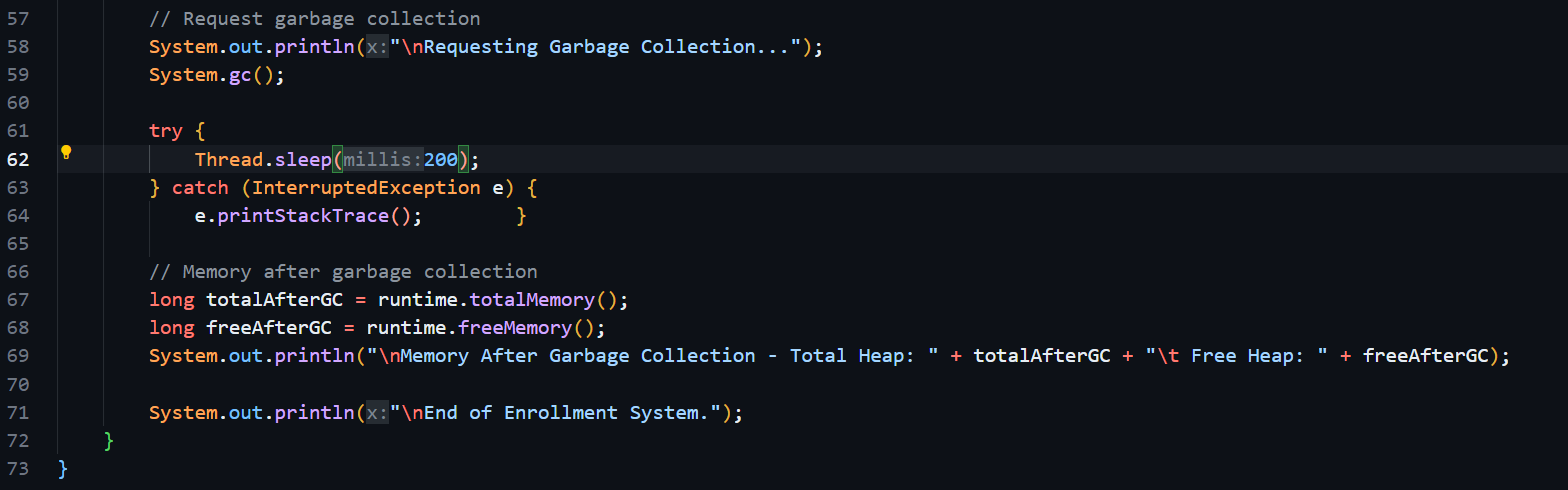
**Output:**

****

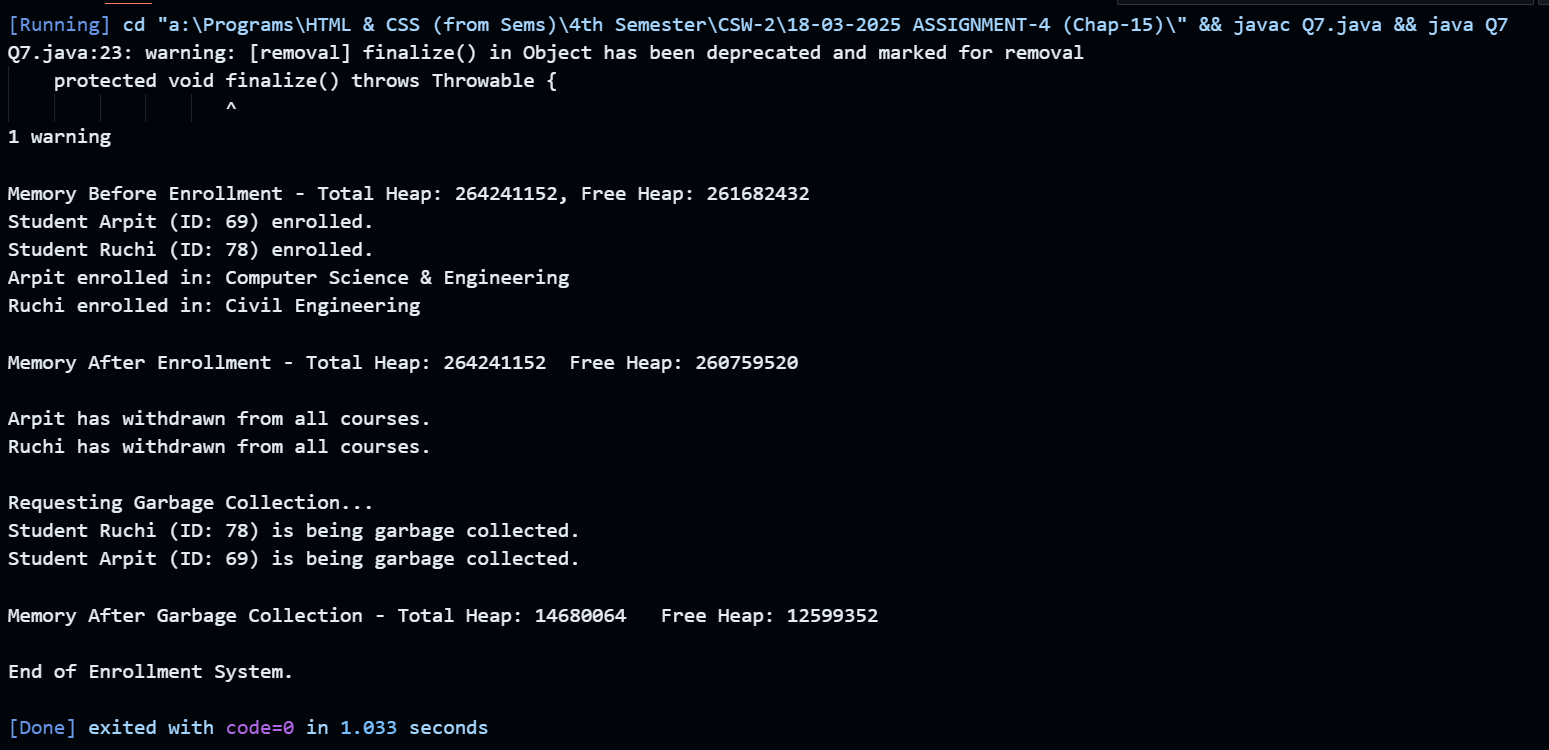
**Q7. Create a Java program for university student enrollment using a Student class to manage course details and student information. Implement efficient garbage collection for memory management and use the Runtime class to monitor memory usage. Override the finalize() method to print a message upon successful garbage collection.**

**Solution: **

****

****

**Output:**

****