**day09\_107856406\_dsdipt\_sudipto\_11june2025**

**Employee Code:** 107856406

**Login ID:** dsdipt

**Email :** dsdipt@amazon.com

**Name:** Sudipto Das

**Date:** 11 June 2025 (Day 09)

### ***Task 1: What do you understand by exceptions?***

An **exception** in programming is an **unexpected or abnormal condition** that occurs during the execution of a program, which **disrupts the normal flow** of instructions.

### ***Task 2: What are the types of exceptions in Java?***

**Checked Exceptions**

* **Checked at compile-time**.
* Must be handled using try-catch or declared with throws.
* **Examples**:
  + IOException
  + SQLException
  + FileNotFoundException

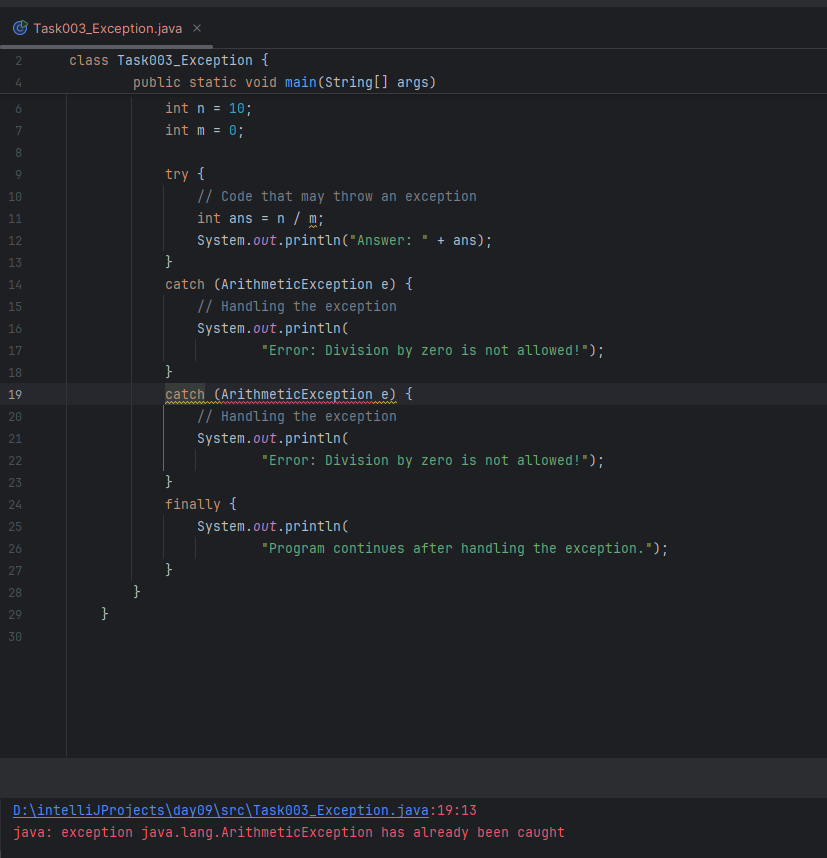
**Unchecked Exceptions**

* **Occur at runtime**.
* Not checked by the compiler.
* Caused by programming logic errors.
* **Examples**:
  + NullPointerException
  + ArithmeticException
  + ArrayIndexOutOfBoundsException

**Errors**

* Represent **serious system-level issues**.
* Usually **cannot be handled** by the program.
* **Examples**:
  + OutOfMemoryError
  + StackOverflowError
  + VirtualMachineError

### ***Task 3: try the below code and tell which kind of exception it is? What is the output?***



**Output of the Program:**

Error: Division by zero is not allowed!

Program continues after handling the exception.

### ***Task 4: List of checked and unchecked exceptions.***

✅ **Checked Exceptions (Compile-Time Exceptions)**

These exceptions are checked by the compiler at compile-time. If not handled properly using try-catch or declared with throws, the program will not compile.

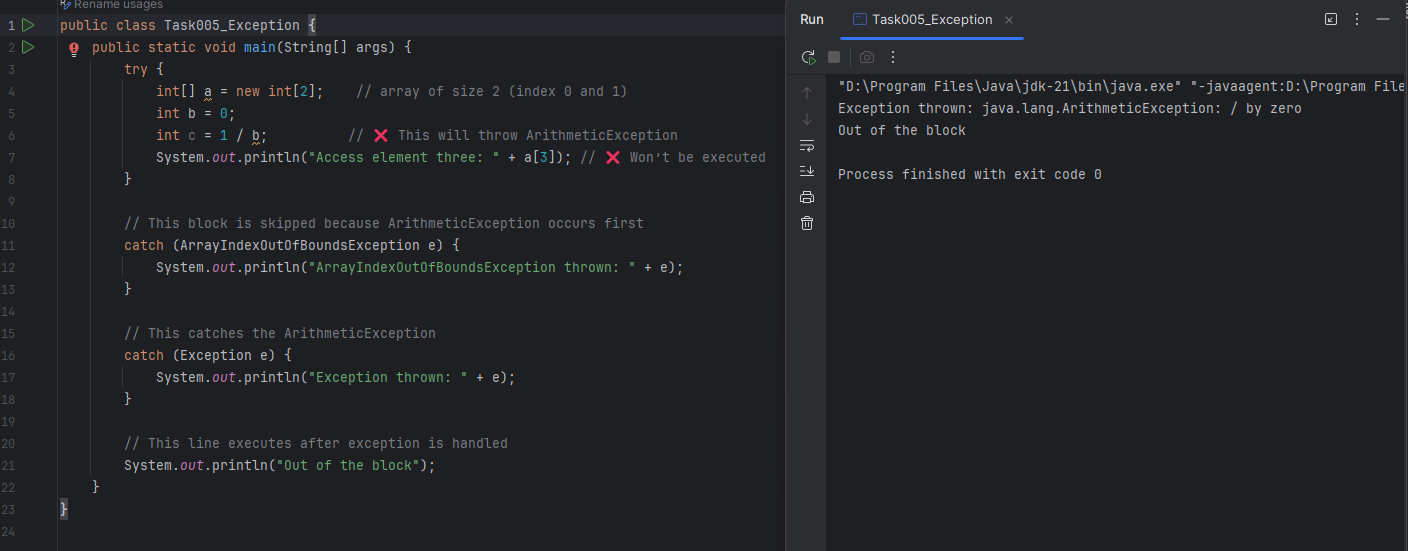
* **IOException** – Occurs during input/output operations, like reading from a file.
* **FileNotFoundException** – Thrown when the specified file cannot be found.
* **ClassNotFoundException** – Happens when a class cannot be located at runtime.
* **SQLException** – Related to database access errors.
* **InterruptedException** – Thrown when a thread is interrupted during execution.
* **ParseException** – Occurs when parsing formatted input like dates or numbers fails.
* **NoSuchMethodException** – Raised when a method that is being called doesn't exist.
* **InstantiationException** – Thrown when trying to create an object of an abstract class or interface.
* **InvocationTargetException** – Wraps exceptions thrown by methods invoked using reflection.

⚠️ **Unchecked Exceptions (Runtime Exceptions)**

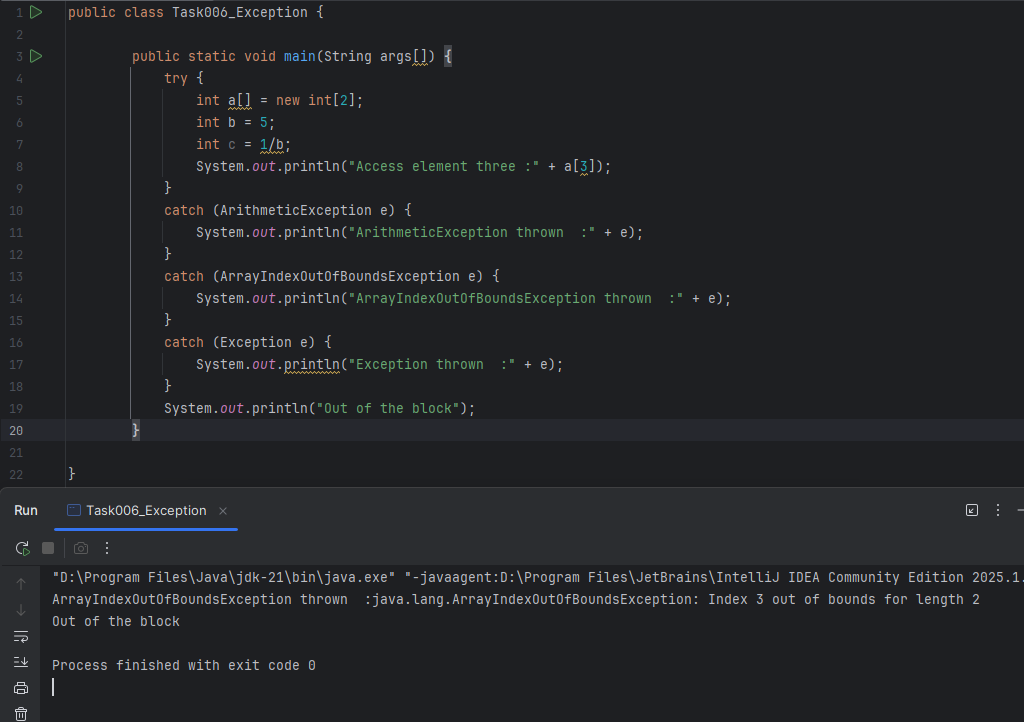
These are not checked at compile-time. They usually indicate bugs in the code and occur during program execution.

* **ArithmeticException** – Thrown when an illegal arithmetic operation occurs, like division by zero.
* **NullPointerException** – Occurs when calling methods on a null object reference.
* **ArrayIndexOutOfBoundsException** – Happens when trying to access an array with an invalid index.
* **StringIndexOutOfBoundsException** – Thrown when accessing an invalid index in a string.
* **IllegalArgumentException** – Indicates a method received an inappropriate argument.
* **NumberFormatException** – Raised when trying to convert a string to a number improperly.
* **ClassCastException** – Thrown when an object is cast to an incompatible class type.
* **IllegalStateException** – Indicates that a method has been invoked at an inappropriate time.

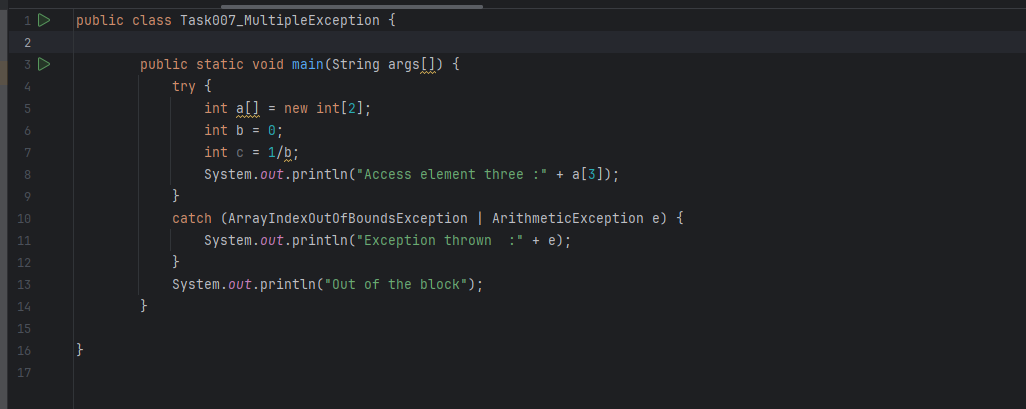
### ***Task 5: Try with Multiple catch blocks, Execute the below code with reason.***



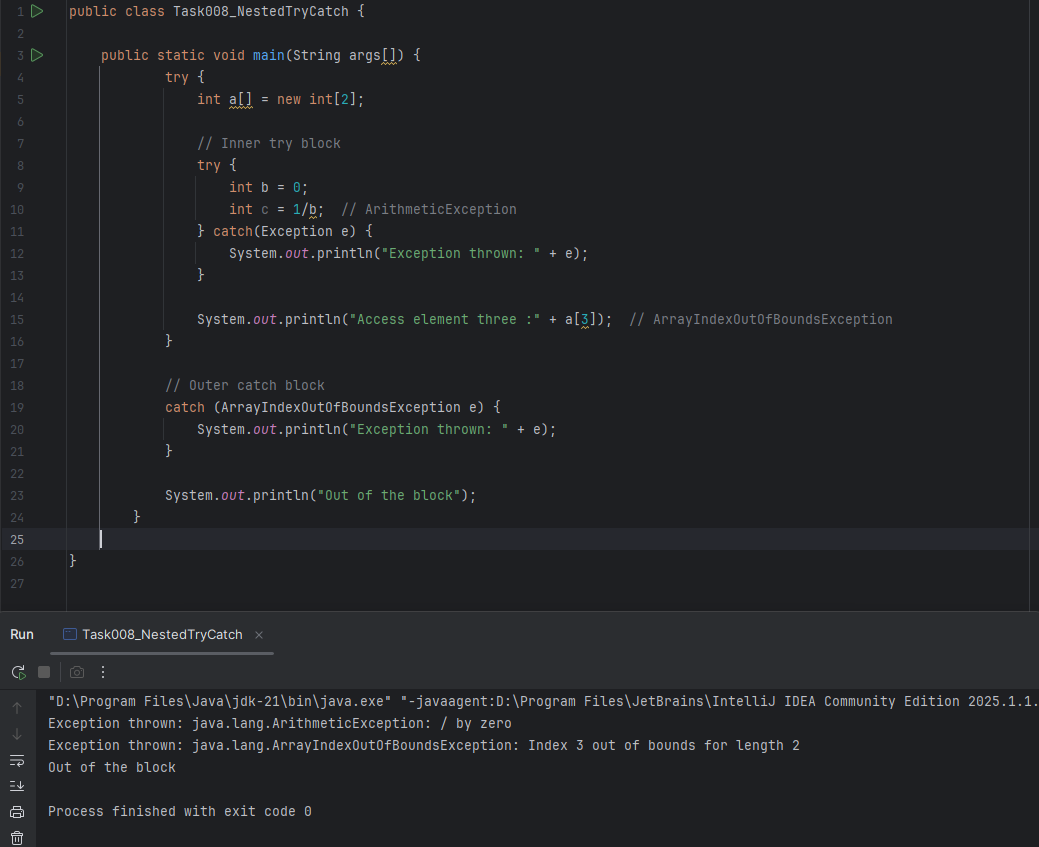
### ***Task 6: What is the output of the below code… give your reason for the output***



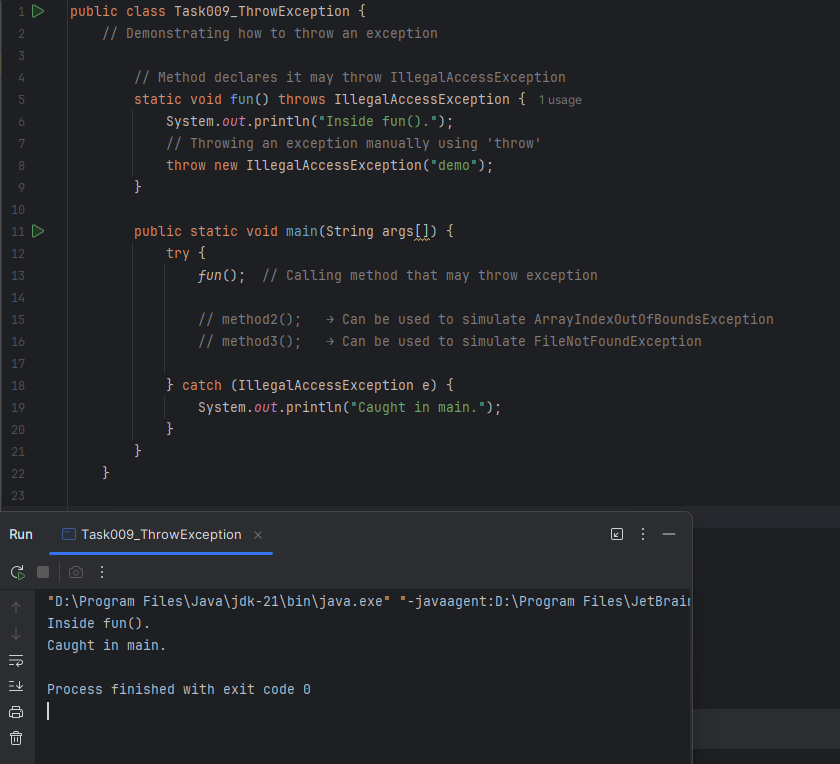
### ***Task 7: Output and Explanation – Multiple Exceptions in a Single Catch***



### ***Task 8: Nested Try-Catch – Output & Explanation***



### ***Task 9: Demonstrating how to throw an exception***

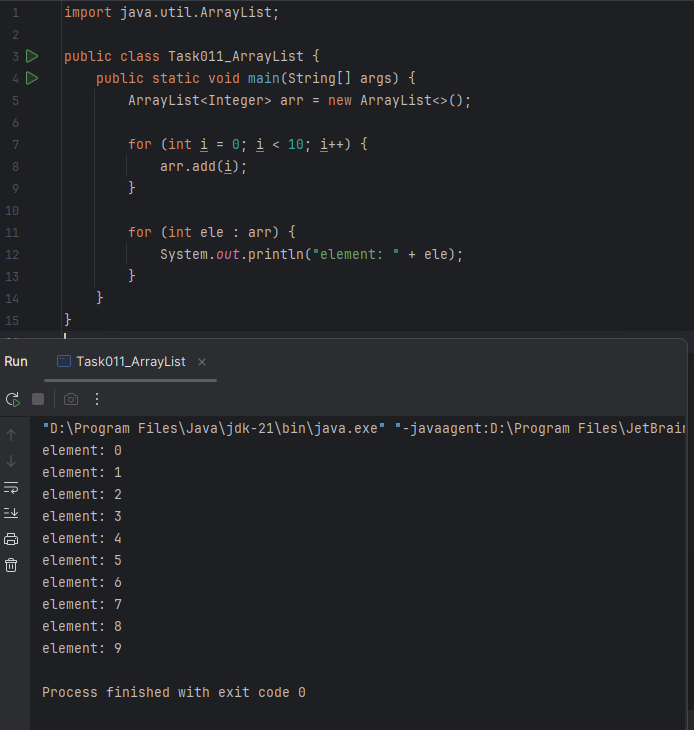


### ***Task 10: Collection Framework***

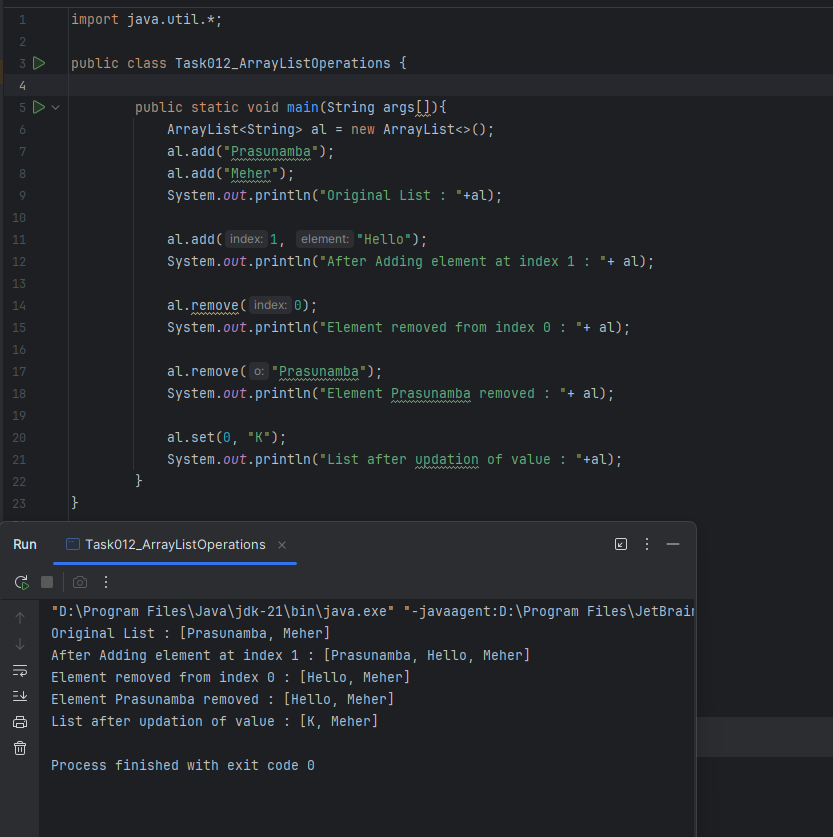
The Java Collection Framework provides **ready-to-use classes and interfaces** for storing and manipulating groups of objects.  
 Common interfaces include:

* List → ordered collection (e.g., ArrayList, LinkedList)
* Set → unique elements (e.g., HashSet, TreeSet)
* Map → key-value pairs (e.g., HashMap, TreeMap)

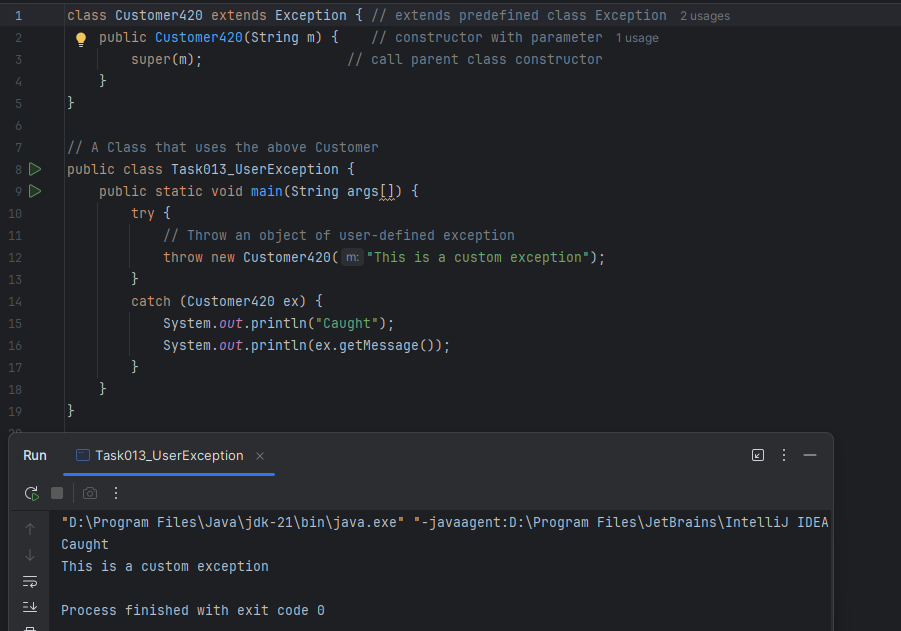
### ***Task 11: Java program to create an ArrayList and display 10 elements using a for loop.***



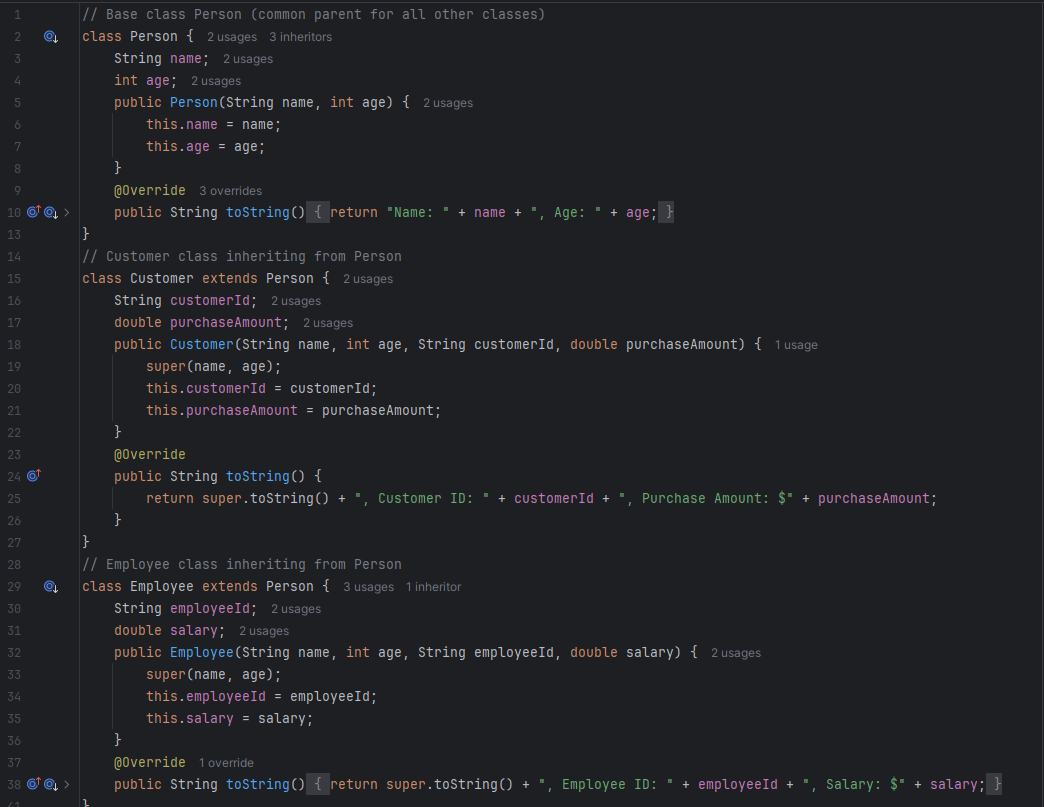
### ***Task 12: ArrayList Operations (Add, Del and Updation of Elements in ArrayList)***

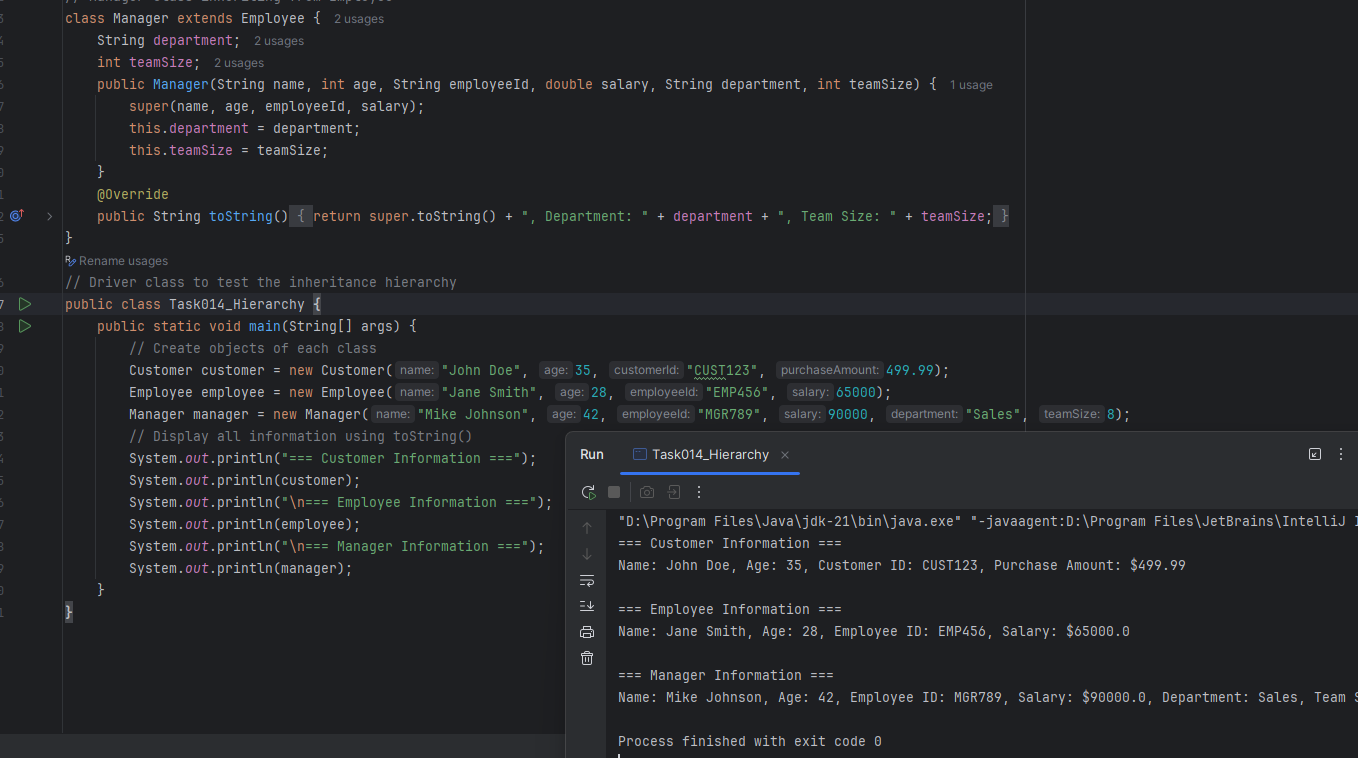


### ***Task 13: User-Defined Exception Example***

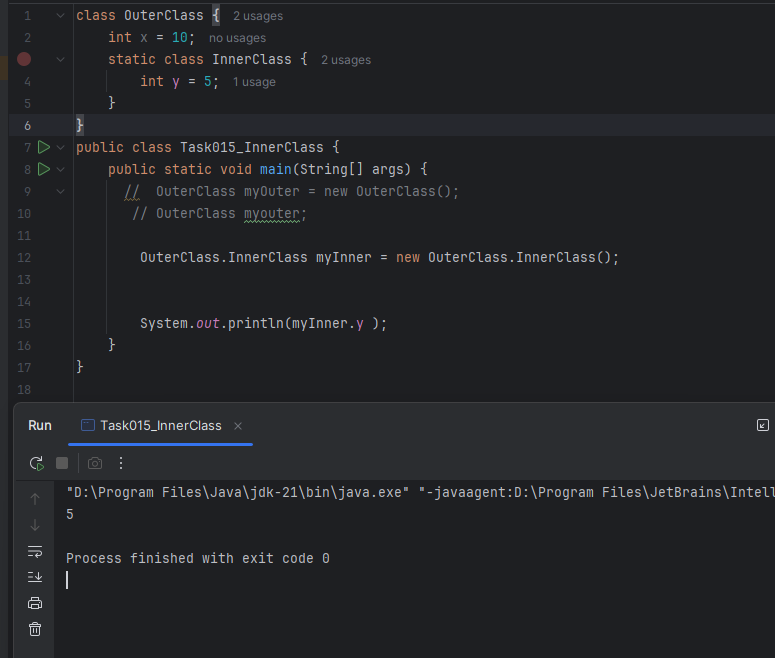


### ***Task 14: Inheritance & Abstraction with Person Hierarchy***

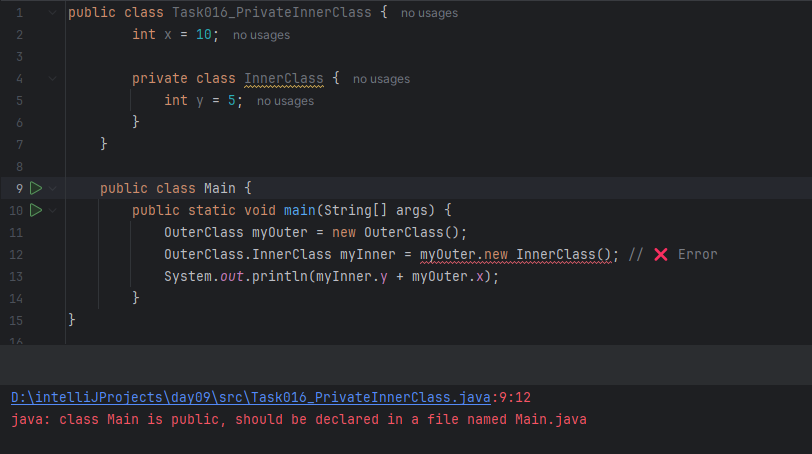




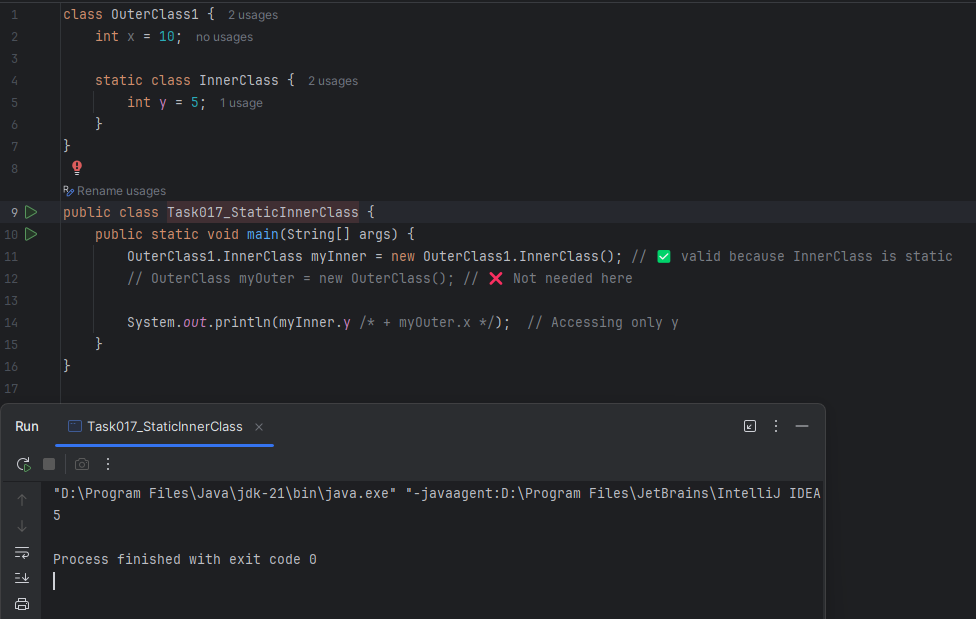
### ***Task 15: Output and Explanation – Inner Class Example***



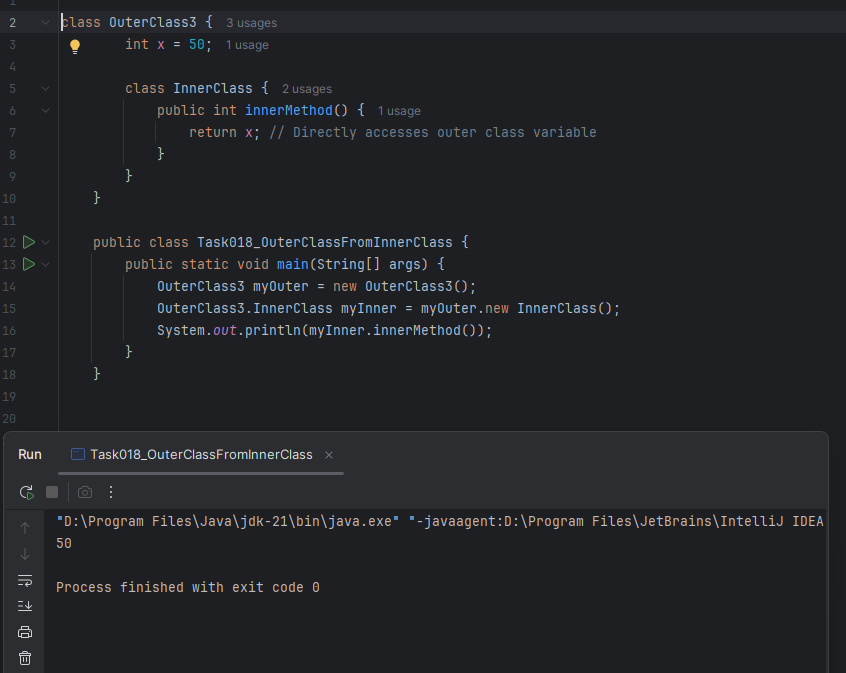
### ***Task 16: Making the Inner Class private – Output & Explanation***



### ***Task 17: Make Inner Class static – Output & Explanation***



### ***Task 18: Access Outer Class Variable from Inner Class Method***



### 