Q1. There are four accesses modifiers in Java 🡪 **Public, Private, Protected and Default**

Their significance in term of class, method and variable accessibility is as follows

**Public:**

Classes, methods, and variables declared as public are accessible from any other class. There are no restrictions on accessing public elements from outside the package or program.

**Private:**

Methods and variables declared as private are accessible only within the same class. This is commonly used to hide the internal implementation details of a class from outside access.

**Protected:**

Methods and variables declared as protected are accessible within the same package or by subclasses in different packages. This allows for restricted access within the package and is also accessible to subclasses outside the package.

**Default:**

When no access modifier is specified, it is considered as default. Classes, methods, and variables with default access are accessible only within the same package. This is useful for creating classes and methods that are only intended for use within a particular package.

Q2)

**Exceptions:**

Exceptions represent exceptional conditions that occur during the execution of a program. These conditions typically arise due to user error, faulty code, or unexpected situations that disrupt the normal flow of execution. Exceptions can be handled using try-catch block or throwing exceptions and the program can continue its execution.

Example: NullPointerException, ArrayIndexOutOfBoundsException, and IOException

**Errors:**

Errors, on the other hand, represent serious problems that are beyond the control of the application. These are typically caused by the environment in which the application is running or by fundamental resource constraints. Errors are not recoverable and often indicate severe problems that require intervention at a higher level, such as by system administrators or developers.

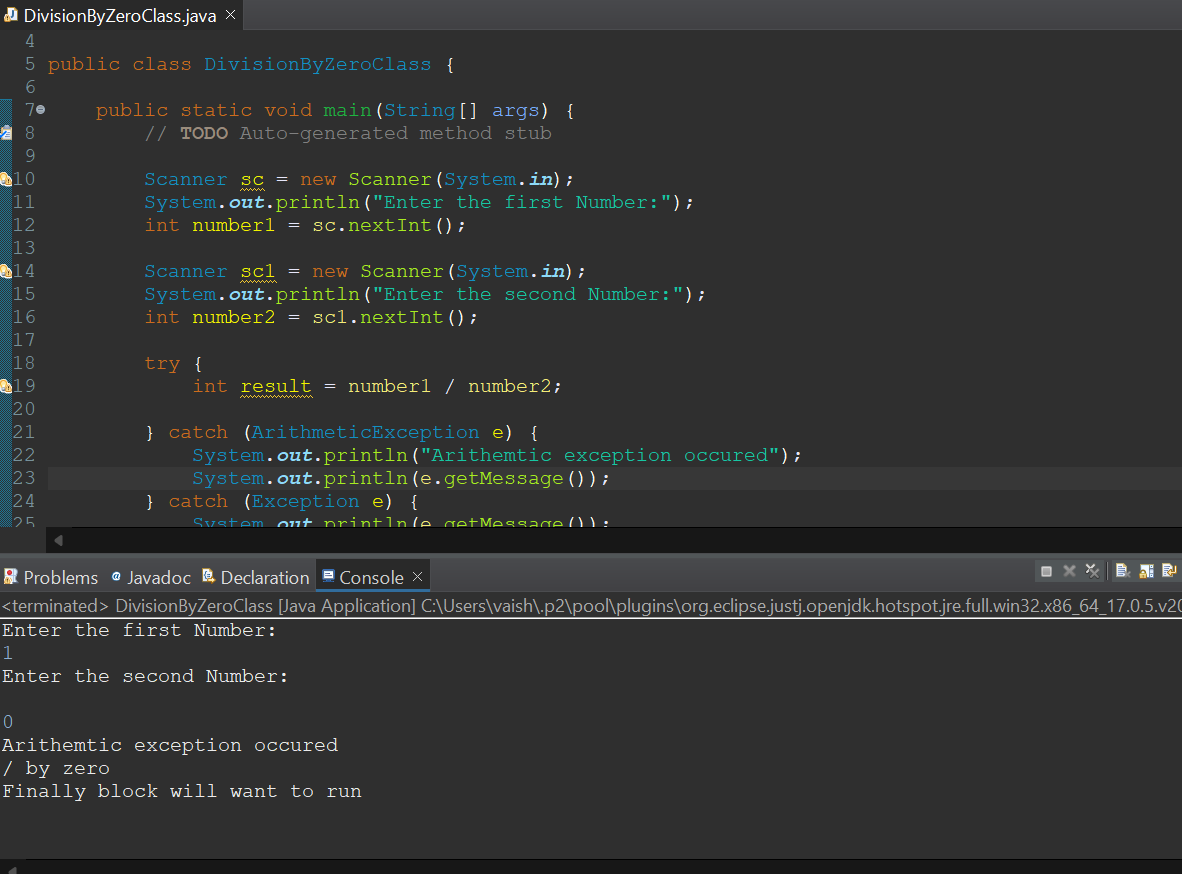
Example: OutOfMemoryError, StackOverflowError

**Q3)**

| Checked Exception | Unchecked Exception |
| --- | --- |
| Checked exceptions occur during compile time when the source code is being converted into an executable code. | Unchecked exceptions occur during runtime when the program is in execution. |
| The compiler checks the checked exception. | The compiler does not check these types of exceptions. |
| This type of exception is considered as a subclass of the exception class. | Unchecked exceptions occur during runtime, therefore, they are not included in the exception class. |
| The Java Virtual Machine requires the checked exception to be caught or handled. | The Java Virtual Machine does not require the unchecked exception to be caught or handled. |
| Example: IOException, SQLException | Example: Null pointer Exception, ArrayIndexOutofBounds Exception |

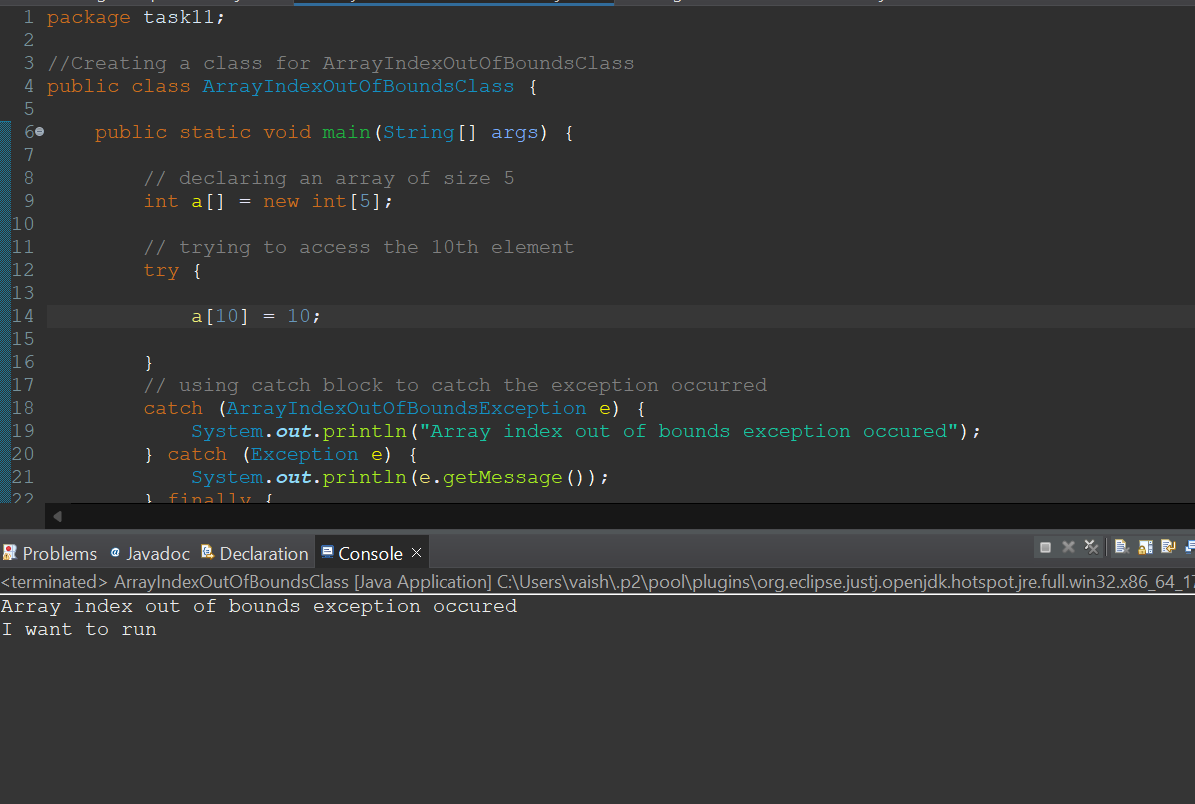
Q4)

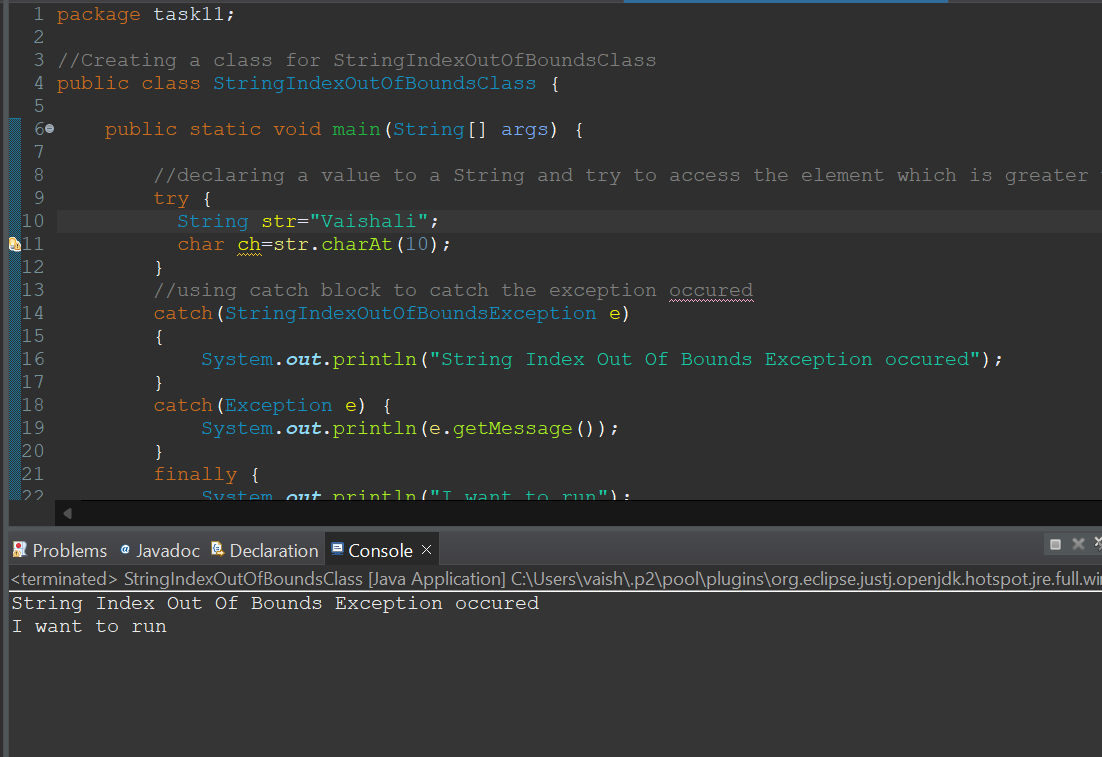
Output:



**Q5)**

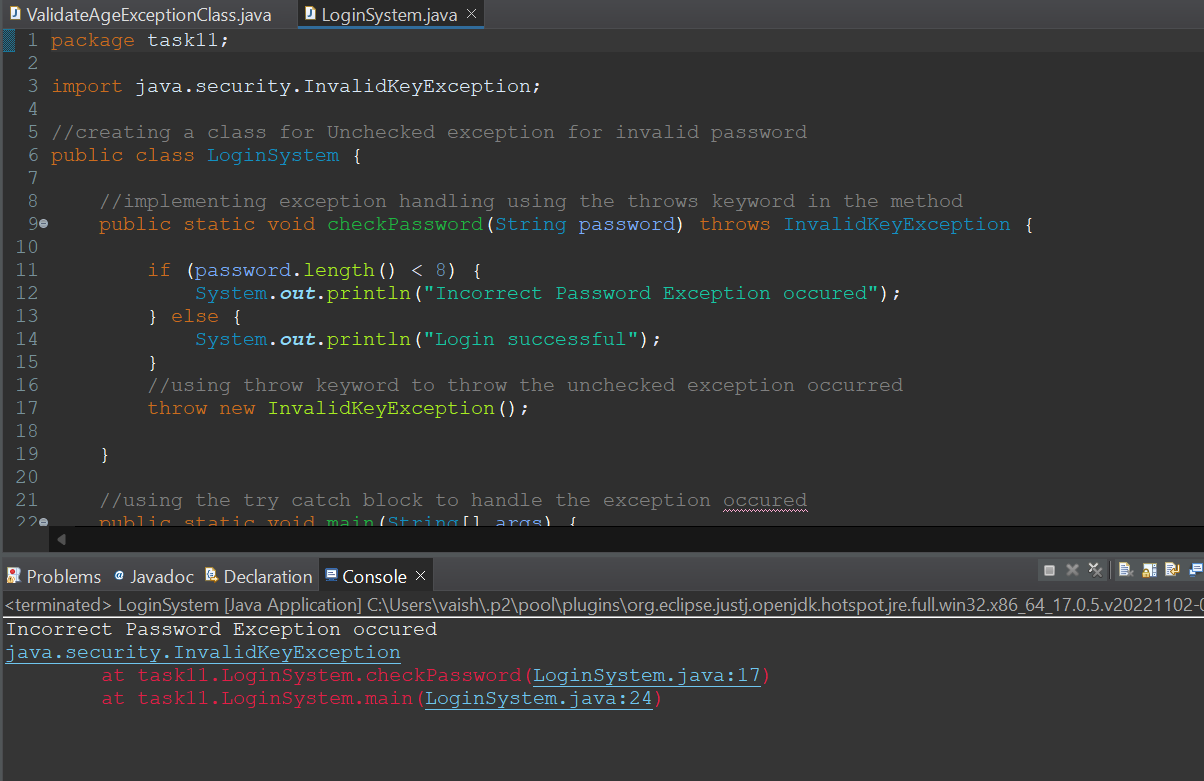
**Output:**

****

****

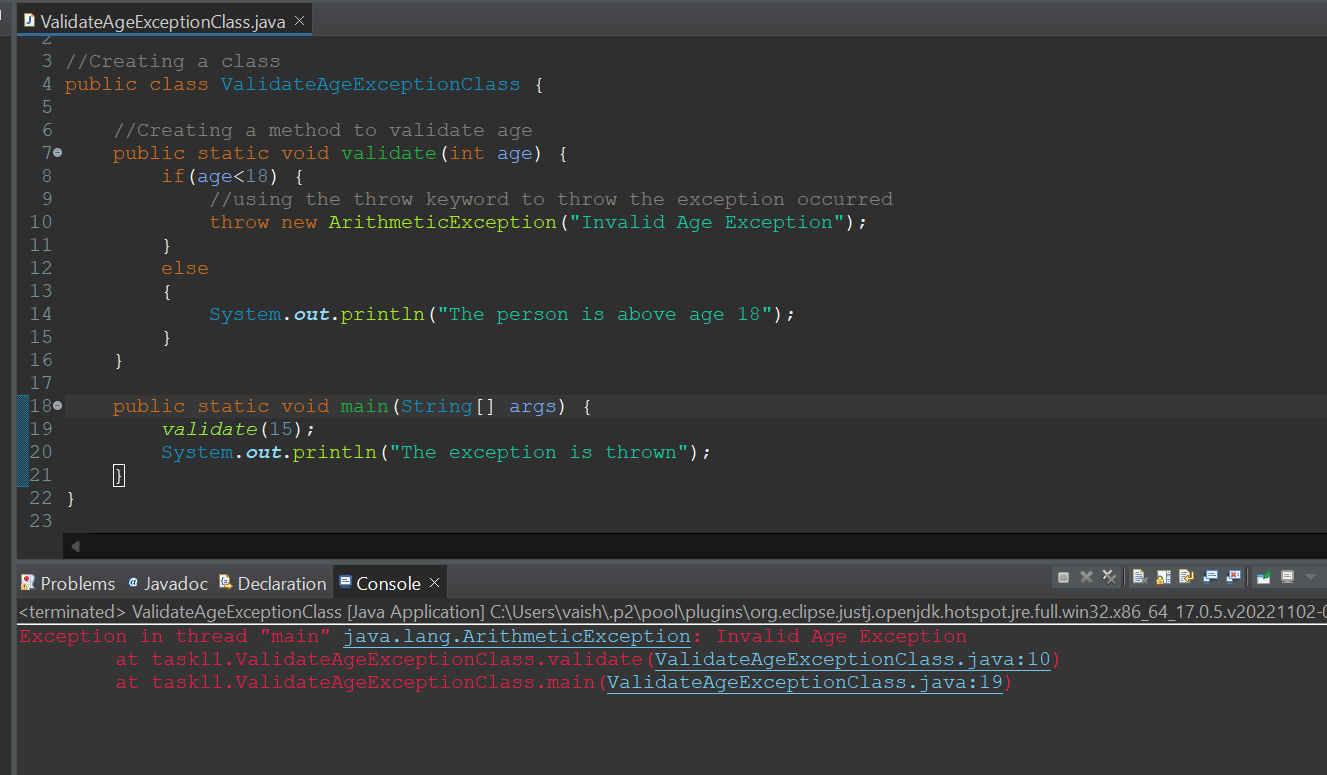
**Q6)**

**Output:**

****

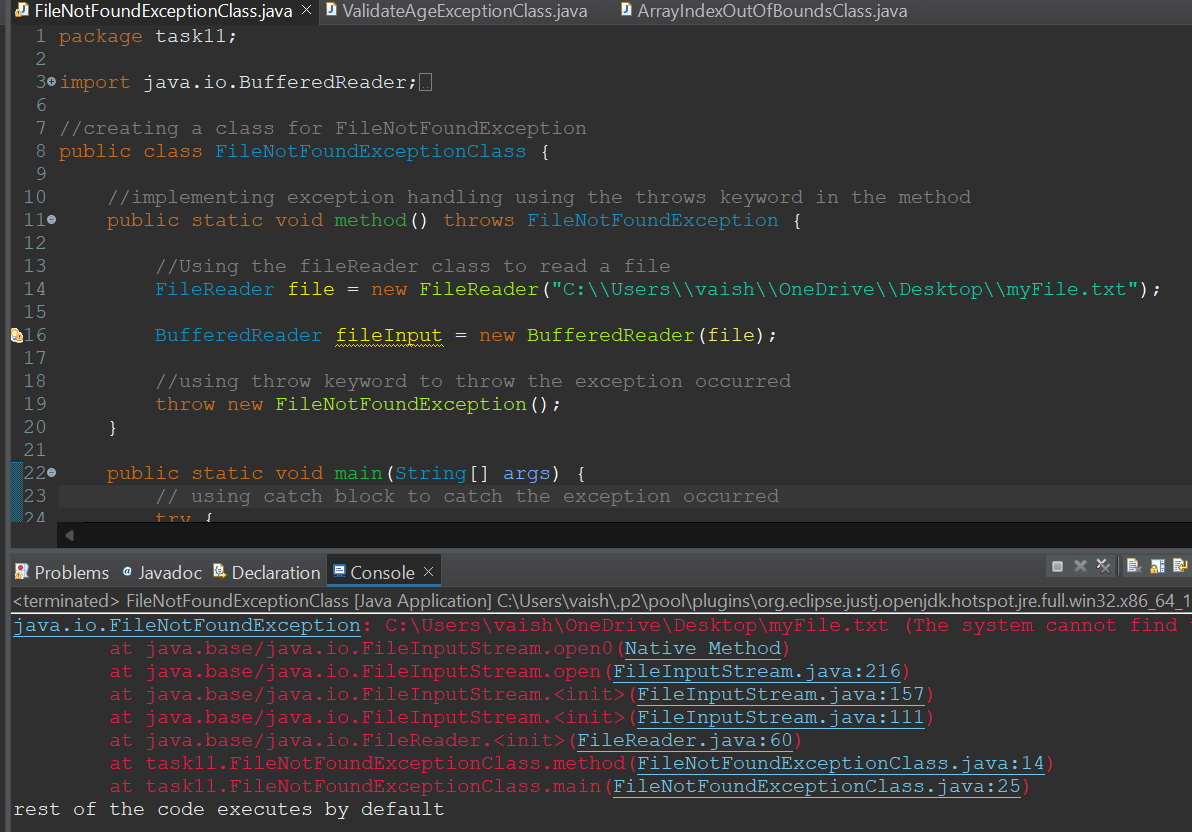
**Q7)**

**Output:**

****

**Q8)**

**Output**

****