Practical session 6

This work should be completed before the next lecture.

Task 1: A generic `Stack` class with custom exceptions

Copy your NetBeans project from Task 3 of Week 5.

Modify the generic `Stack` class as follows:

- Make the collection of objects an array with 5 places instead of an `ArrayList` or an `ArrayDeque`
- Throw a `FullStackException` custom exception when `push()` is called on a full stack
- Throw an `EmptyStackException` custom exception when `pop()` is called on an empty stack

In the `main()` method of the application:

- Create a `String` version and a `Person` version of the `Stack` class
- Push six items to each stack and ensure the `FullStackException` custom exception is thrown
- Pop six attempts item from each stack and ensure the `EmptyStackException` custom exception is thrown
- Ensure the output to the console shows all push and pop attempts and reports all errors

Portfolio requirements:

- The NetBeans project for this completed task

Task 2: An auto-closeable generic `Stack` class

Copy your NetBeans project from Task 1.

Make the generic `Stack` class an auto-closeable resource. In the `close()` method, empty the stack.

In the `main()` method of the application:

- Use one `try`-with-resources statement to open and close both versions of the `Stack` class
- Push three items onto each stack inside the `try`-with-resources statement

Portfolio requirements:

- The NetBeans project for this completed task

Task 3: Team and Player using `try`-with-resources statements

Copy your NetBeans project from Task 1 of Week 4.

Modify the code so use auto-closeable resources in `try`-with-resources statements.

Portfolio requirements:

- The NetBeans project for this completed task
- The binary data file created by the program

Task 4: Catch up

Use the rest of your time this week to catch up on any tasks that you have not yet completed from all previous weeks.