

# CSC 243, Spring 2020, Assignment 4

**Purpose:** Generics and Interfaces

**Due:** 11:59pm, Monday, April 6, 2020

## Get the assignment code

These instructions assume that your course git repository is set up. Change into your course repository directory and enter the following commands.

```
git fetch assignments
git checkout assignments/master -- assignment4
git add assignment4
git commit -a
```

This will copy the `assignment4` directory into your working directory, start tracking the files in the `assignment4` directory, and commit those files to your local git repository.

## Assignment Description

In this assignment, you need to make the following changes to the code base:

- Change the `getValidMoves` method to have return type `List<Integer>` instead of `int []`.
- Change the `Board` class to implement the `Cloneable` interface. The `clone` method must return a deep copy.
- Add a new player action `Action.RESET` that allows a player to reset the game to the initial state.
- Add a new player action `Action.UNDO` that allows a player to undo the last action. The undo action has different behavior depending on whether there is an `AIPlayer` involved. In the case of two `ConsolePlayers`, the undo action should cancel the effect of the most recent player action. In the case of one `AIPlayer` and one `ConsolePlayer`, when the `ConsolePlayer` chooses the undo action, the effects of the two most recent player actions should be canceled. The `instanceof` operator can be used to determine which behavior should be used for a given run of the game. One way to implement the undo operation is to save the history of board states in a stack. When the player chooses the undo action, the board state is replaced with the appropriate element from the stack and that state becomes the top of the stack. In order to implement undo in this fashion, you need to get a deep copy of the `Board` object, which means that the `Board` class should be modified.

## Compiling and Running

To compile the programs, execute the command:

```
make
```

To run the main method, execute the command:

```
make run
```

To build the Javadoc documentation, execute the command:

```
make doc
```

To remove the compiled class files, execute the command:

```
make clean
```

## Turning in the Assignment

To turn in the assignment execute the following git commands:

```
git commit -a  
git push origin master
```

Note: the most recent commit before the due date will be considered your official submission.

## Grading Criteria

- Concise, accurate documentation following the CSC Department documentation guidelines
- Correct Javadoc is used for class and method documentation
- Correct implementation of the specification

Note: If your code does not compile, then you will receive a failing grade for this assignment. If the submission includes material that was not covered in class and the material is not properly cited, then you will receive a failing grade for this assignment.