The University of Melbourne
School of Computing and Information Systems

COMP90041

Programming and Software Development

Second Semester, 2018

Fourth Assessed Exercise (lab4)

Submission due Saturday, 29 September 2018, 5:00PM

This exercise is to be assessed, and so **must be done by you alone**. Sophisticated similarity checking software will be used to look for students whose submissions are similar to one another.

This week you will implement the `BackgammonBoard` class you wrote tests for in the previous assessed lab. That is, you should write a Java class `BackgammonBoard` that represents a Backgammon board. It should implement the following methods and constructors:

`BackgammonBoard()` (constructor) Creates a fresh `BackgammonBoard` with no men on it.

`int getPointCount(int point)` Returns the number of men on the point.

`boolean getPointBlack(int point)` Returns `true` if the men on the specified point are black, or `false` if they are white. If there are no men on that point, the result may be either `true` or `false`.

`void setPoint(int point, int count, boolean black)` Sets the number of men on the specified point to the specified count, and sets their colour to black if `black` is true, or white if it is false.

`int getBarBlackCount()` Returns the number of black men on the bar.

`int getBarWhiteCount()` Returns the number of white men on the bar.

`void move(int fromPoint, int toPoint)` Moves one men from the specified `fromPoint` to the specified `toPoint`, if the move is legal; if it is illegal, this method does nothing.

For `setPoint` and `move` methods, if a specified point number is not between 0 and 23, the method should do nothing. For `getPointCount` and `getPointBlack`, if the specified point is out of bounds, the value returned may be any valid value of the appropriate type.

Hint: Arrays are a good way to store the number and colour of men on each point.

Hint: You should use your `BackgammonTest` class from the last assessed lab to test your class. You do not need to submit your testing code, only the `BackgammonBoard` class. You will need to add extra tests, to ensure that all the methods behave correctly, not just `move`, and that your methods do not crash (with an exception) on invalid input.

Submission

You must submit your project from any one of the student unix servers. Make sure the version of your program source files you wish to submit is on these machines (your files are shared between all of them, so any one will do), then `cd` to the directory holding your source code and issue the command:

```
submit COMP90041 lab4 BackgammonBoard.java
```

Important: you must wait a minute or two (or more if the servers are busy) after submitting, and then issue the command

```
verify COMP90041 lab4 | less
```

This will show you the test results and the marks from your submission, as well as the file(s) you submitted. If the test results show any problems, correct them and submit again. You may submit as often as you like; only your final submission will be assessed.

If you wish to (re-)submit after the project deadline, you may do so by adding "`.late`" to the end of the project name (*i.e.,* `lab4.late`) in the `submit` and `verify` commands. But note that a penalty, described below, will apply to late submissions, so you should weigh the points you will lose for a late submission against the points you expect to gain by revising your program and submitting again. **It is your responsibility to verify your submission.**

Late Penalties

Late submissions will incur a penalty of 1% of the possible value of that submission per hour late, including evening and weekend hours. This means that a perfect project that is a little more than 2 days late will lose half the marks. These lab exercises are frequent and of low point value, and your lowest lab mark will be dropped. Except in unusual circumstances, I will not grant extensions for lab submissions.

Academic Honesty

This lab submission is part of your final assessment, so cheating is not acceptable. Any form of material exchange between students, whether written, electronic or any other medium, is considered cheating, and so is the soliciting of help from electronic newsgroups. Providing undue assistance is considered as serious as receiving it, and in the case of similarities that indicate exchange of more than basic ideas, formal disciplinary action will be taken for all involved parties.