

## Introduction

This is Coursework assignment 1 (of 2 coursework assignments in total) for 2016–2017. Part (a) asks that you demonstrate an understanding of simple graphics, inner classes, events and the *ActionListener* interface. Part (b) looks at the object-oriented programming paradigm: inheritance, interfaces, classes (including abstract classes) and instance methods.

**IMPORTANT NOTE:** Please use the most recent version of Java, Java 8.

## Electronic files you should have:

### Part (a)

- *RectanglesGUI.java*

### Part (b)

- *BaseQuiz.java*
- *FreeQuiz.java*
- *MultipleChoiceQuiz.java* – *getQuestions* method only
- *TrueFalseQuiz.java* – *getQuestions* method only
- *Question.java*
- *FreeQuizQuestion.java*
- *MultipleChoiceQuestion.java*
- *GenericQuizClasses.pdf*

## What you should hand in: very important

At the end of each section there is a list of files to be handed in – **please note the handing in requirements supersede the generic University of London instructions.** Please ensure that you hand in **electronic versions** of your .java files since you cannot gain any marks without handing them in. Class files are **not** needed, and any student handing in only a class file will **not** receive any marks for that part of the coursework assignment, **so please be careful about what you upload as you could fail if you submit incorrectly.**

There is one mark allocated for handing in uncompressed files – that is, students who hand in zipped or .tar files or any other form of compressed files can only score 49/50 marks.

There is a further one mark allocated for handing in files that are **not** contained in a directory; students who upload their files in a directory can only achieve 49/50 marks.

Please put your name and student number in the form of a comment at the top of each .java file that you hand in.

**The examiner intends to compile and run your Java programs; for this reason, programs that will not compile will not receive any marks.**

You are asked to give your classes certain names; please follow these instructions very carefully. Make sure there is no conflict between your file name and your class name. Remember that if you give your files names that are different from the names of your classes (e.g. *cwk1-partA.java* file name versus *RectanglesGUI* class name) your program will not compile because of the conflict between the file name and the class name.

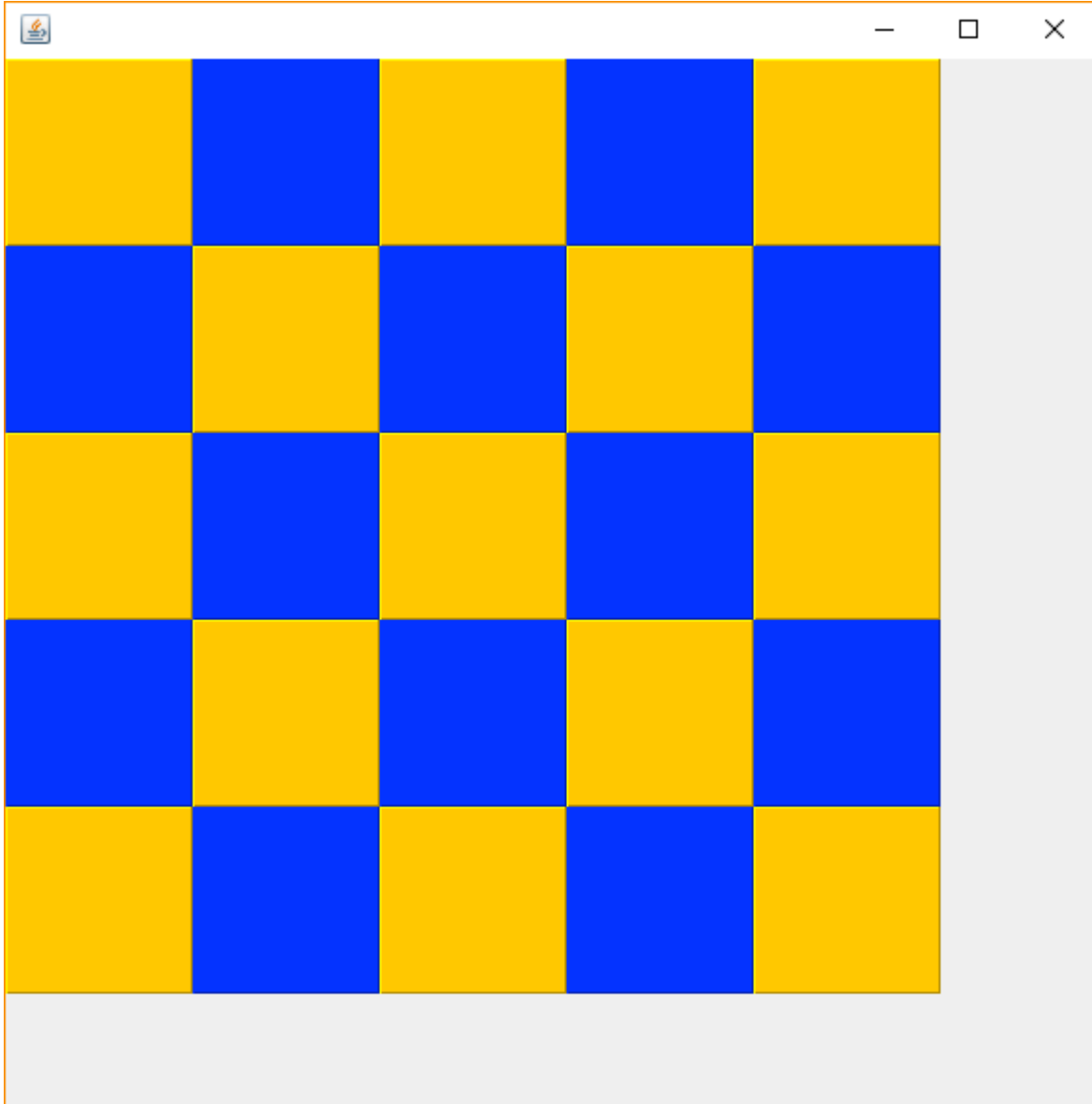
Students who hand in files containing their Java classes that cannot be compiled (e.g. PDFs) will not be given any marks for that part of the coursework assignment.

## CO2220 Coursework assignment 1

### Part (a)

Compile and run the *RectanglesGUI* class. You should note that it uses an inner class to place a *JPanel* onto a *JFrame*. Some coloured rectangles are drawn on the *JPanel*:

#### Output of the *RectanglesGUI* class



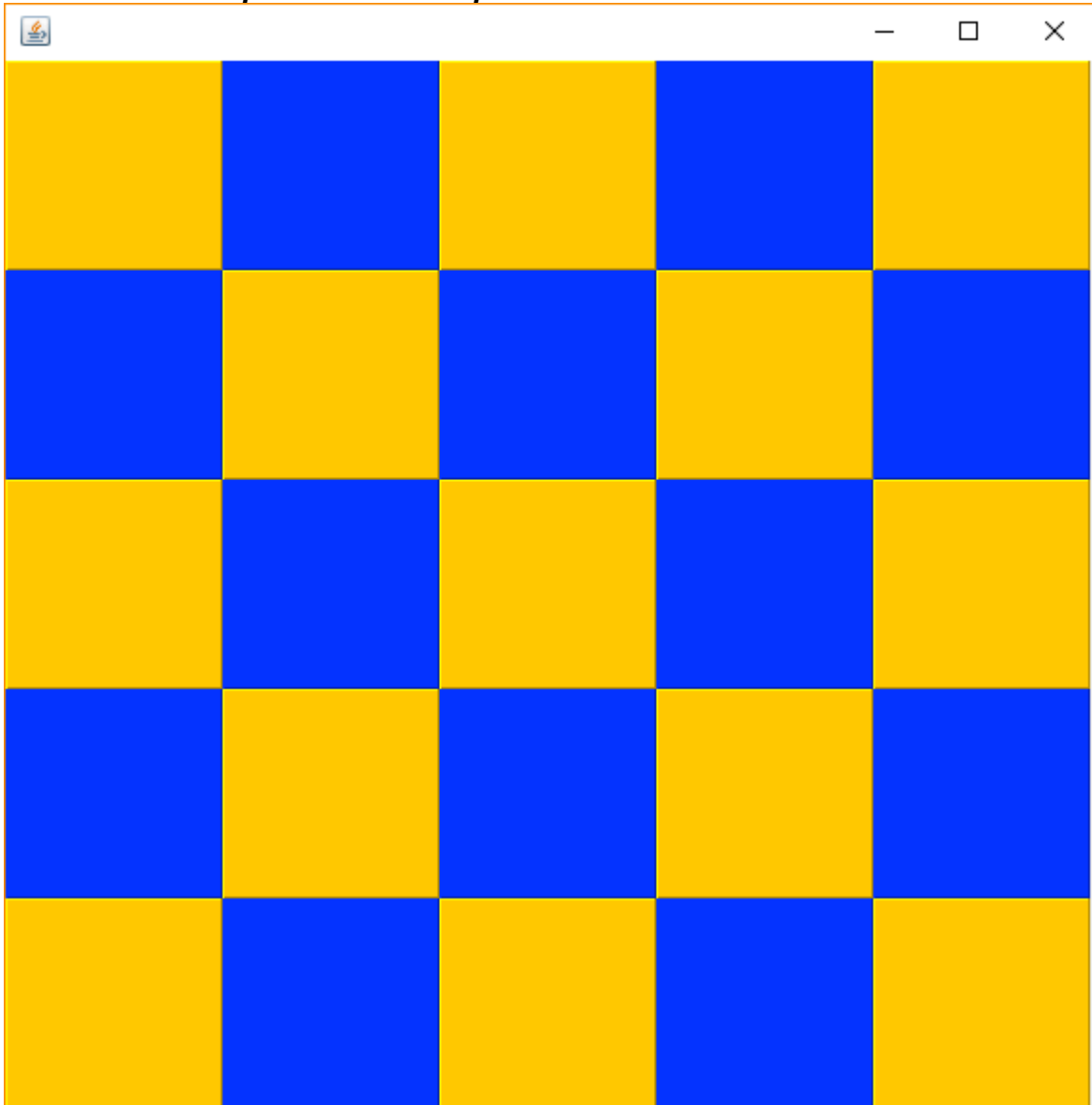
Please complete the following tasks:

1. Shorten the *paintComponent (Graphics)* method by rewriting the statements to draw and fill the rectangles with colour. Put the statements into a loop of your choosing. The model answer uses nested *for* loops, and has reduced the number of statements needed to draw the rectangles to 5.

[5 marks]

2. Using *getWidth()* and *getHeight()* for the *RectangleDrawPanel*, distribute the rectangles evenly across the *drawPanel*, so that each is fully visible and so that all of the available space is used (see the image below).  
  
NOTE: your solution must be scalable – that is, the dimensions of the *JFrame* could be changed to be bigger or smaller, and the rectangles would still all be visible and fill the available space, without any other changes being made. [3 marks]
3. Add a *JButton* to the *SOUTH* region of the *JFrame* using *BorderLayout* (2 marks). Set the text on the button. The text should invite the user to click the button (1 mark). [3 marks]
4. Write an inner class called *RandomColorListener* to implement the *ActionListener* interface and to listen to the *JButton* in the *SOUTH* region.  
  
NOTE: you can only get full credit for this part of the question by writing an inner class. [3 marks]
5. When the user clicks on the *JButton* in the *SOUTH* region, the rectangles filled with *color1* should all change to a random *Color*, while the rectangles filled with *color2* should not change. A second click on the *JButton* should make the rectangles filled with *color2* all change to a random *Color*, while the rectangles filled with a random *Color* by the first click should stay the same *Color*. The user should be able to continue clicking on the button indefinitely, and with each click one set of rectangles will be filled with a random *Color*. In each case, the rectangles to change should be the ones that stayed the same on the last click (in other words, the *Color* change should alternate between the two sets of rectangles). This means that with each click only one set of rectangles should change colour. [3 marks]
6. Write a second inner class implementation of the *ActionListener* interface, called *ResetListener*. The *ActionPerformed(ActionEvent)* method of the class should have a means of resetting the colours of the rectangles back to the colours that they were filled with at the start (orange and blue). [3 marks]
7. Add a second *JButton* to the *NORTH* region, and add the *ResetListener* to it such that when the button is clicked the rectangles are once again filled with orange and blue, as they were at the start of the program. Further clicks on the button will have no effect when the default orange and blue colours are displayed, but will reset the colours to the default setting once more if they have been changed by clicking the other button (2 marks). Add an appropriate message to the button telling the user what it does (1 mark). [3 marks]

**Output of the RectanglesGUI class, revised so that the rectangles are all the same size and divide up the available space between them.**



### Reading for Part (a)

- See: <http://docs.oracle.com/javase/7/docs/api/java/awt/Color.html> for the constructors, for example `Color(int r, int g, int b)`, that allow developers to make their own *Colors*.
- Chapter 11 of the Subject Guide, pages 115–17.
- Chapter 12 of volume one the Subject Guide, pages 131–32.
- Chapter 12 of *Head First Java*, pages 353–81.

### Deliverable for part (a)

- An electronic copy of your revised program: *RectanglesGUI.java*

*Please put your name and student number in the form of a comment at the top of your file.*

## Part (b)

In this part of the coursework assignment, you have been given the *BaseQuiz* and the *FreeQuiz* classes, together with the *Question*, *FreeQuizQuestion* and *MultipleChoiceQuestion* classes. *BaseQuiz* is abstract, and *FreeQuiz* is one possible concrete class extension of it. You should compile and run the *FreeQuiz* class and see how it behaves. The *FreeQuiz* class uses *FreeQuizQuestion* objects, which are placed into an *ArrayList* of potential questions. The number of possible questions to be randomly chosen from this *ArrayList* and asked of the user is set in the *getQuizSize()* method. You might want to try changing the value (currently 5) a few times to see what happens.

The *FreeQuizQuestion* class is a child class of the *Question* class. The *Question* class specifies that the question itself will be a *String*, but it does not set the type of answer, which is left to the child classes. *FreeQuizQuestion* answers are *Strings*, while the *MultipleChoiceQuestion* class answers are more complicated. Multiple choice questions need a list of possible answers, and a record of the correct answer. In the *MultipleChoiceQuestion* class the possible answers are stored in an array, and the actual answer is recorded as an *int* that holds the position of the correct answer in the list of possible answers; the list of possible answers are numbered starting at 1. Hence in the question:

```
MultipleChoiceQuestion q1 = new MultipleChoiceQuestion("What  
is the primitive data type that starts with 'i'?", 1, "int",  
"INT", "Int", "Integer");
```

the '1' means that "int" is the correct answer.

You have been given the *MultipleChoiceQuiz* and *TrueFalseQuiz* classes; each contains only the *populateList()* method.

You have been given a PDF with some *Generic* classes. These classes, *GenericBaseQuiz*, *GenericQuestion*, *GenericFreeQuiz* and *GenericFreeQuizQuestion* have been adapted from the *BaseQuiz*, *Question*, *FreeQuiz* and *FreeQuizQuestion* classes respectively.

Please complete the following tasks:

1. Complete the *MultipleChoiceQuiz* class as a direct child class of *BaseQuiz*. Include a main method in your class with test statements to make and run an instance of the class. Your completed class should behave in the same way as the *FreeQuiz* class, except that the questions that it is asking are multiple-choice ones. This means that the question is displayed with a numbered list of possible answers, and the user has to type in the number of the answer that they think is correct. See the Appendix for an example of the output of this class. [7 marks]
2. Write the *TrueFalseQuestion* class as a direct sub-class of the *Question* class. [6 marks]

3. Complete the *TrueFalseQuiz* class as a direct child class of *BaseQuiz*. Include a main method in your class with test statements to make and run an instance of the class. Your completed class should behave in the same way as the *FreeQuiz* class, except that the questions that it is asking are true or false. This means that the user is shown a statement and has to decide whether they think the statement is true or false, typing in their answer. See the Appendix for an example of the output of this class. [7 marks]
4. Can you explain how and why the *Generic* classes are different from the classes that they have been based on? (2 marks)

Consider the *getRandomQuestions(List, int)* method in the *GenericBaseQuiz* class. The corresponding method was static in *BaseQuiz*, but is an instance method in *GenericBaseQuiz*. Can you explain why? (3 marks)

You should write no more than two paragraphs in answer to this question (note a paragraph is at most 8 sentences). You may give in your answer as a PDF, Word, OpenOffice or text file. [5 marks]

### Reading for Part (b)

The following chapters and pages of Volume 1 of the Subject Guide, and associated *Head First Java* readings:

- Chapter 3 (Object programming)
- Chapter 5, pages 39–41 (Object behaviour)
- Chapter 7, pages 63–65 (The Java library)
- Chapter 8, pages 73–75 (Inheritance)
- Chapter 9, sections 9.1–9.4 (Abstraction)

Reading for Question 4:

- <https://docs.oracle.com/javase/tutorial/java/generics/>

### Deliverables for Part (b)

*Please put your name and student number in the form of a comment at the top of your Java files.*

Please hand in an electronic copy of the following:

- Your completed *MultipleChoiceQuiz.java*
- *TrueFalseQuestion.java*
- Your completed *TrueFalseQuiz.java*
- A PDF, Word, OpenOffice or text file with your answer to Question 4 Part (b). Please name this file with your surname and student number.

## **MARKS FOR CO2220 COURSEWORK ASSIGNMENT 1**

The marks for each section of Coursework assignment 1 are clearly displayed against each question and add up to 48. There are another two marks available for handing in uncompressed .Java files and for giving in files that are not contained in a directory. This amounts to 50 marks altogether. There are another 50 marks available from Coursework assignment 2.

Total marks for Part (a)	[23 marks]
--------------------------	------------

Total marks for Part (b)	[25 marks]
--------------------------	------------

Mark for giving in uncompressed files	[1 mark]
---------------------------------------	----------

Mark for giving in standalone files; namely, files <b>not</b> enclosed in a directory	[1 mark]
---	----------

<b>Total marks for Coursework assignment 1</b>	<b>[50 marks]</b>
--	-------------------

**[END OF COURSEWORK ASSIGNMENT 1]**



## Appendix

### Sample output of the completed *MultipleChoiceQuiz* class (questions randomly chosen)

```
This quiz has 3 questions. Good luck.
QUESTION (6 possible answers): What is the minimum value of
the byte data type?
1. -63
2. -64
3. -127
4. -128
5. -255
6. -256
Enter a number> 4

QUESTION (3 possible answers): What is the value of the
expression "James".charAt("James".length() - 1) ?
1. no value, run-time error
2. e
3. s
Enter a number> 3

QUESTION (6 possible answers): What is the maximum value of
the byte data type?
1. 63
2. 64
3. 127
4. 128
5. 255
6. 256
Enter a number> 3

You scored 3/3. That's 100%. Excellent!
```

### Sample output of the completed *TrueFalseQuiz* class (questions randomly chosen)

```
This quiz has 5 questions. Good luck.
True or false? Static methods can operate on instance
variables; true
True or false? Instance variables hold the same value for
every instance of the class; true
True or false? Static methods can be run before an instance
of the class is made; true
True or false? Java is case sensitive; true
True or false? The following expression type checks: int x =
"elf".compareTo("7")+11; false

You scored 2/5. That's 40%. You'll have better days.
```