Brock University

Final Examination, Summer 2017 | **Number of Pages: 1**
Course: COSC 3P91 – D17 | **Number of Students: 26**
Date of Examination: Sep 02, 2017 | **Number of Hours: 2**
Time of Examination: 10:00am-noon | **Instructor: M. Winter**

Instructions

1) **Place all answers in the examination booklet provided.**
2) **You do not have to write `import` statements when using routines from the standard Java libraries.**
3) **You do not have to comment your code.**
4) **The paper totals 40 marks. The marks for each question are indicated on the left (e.g. [10]) and the breakdown within the question in italics (e.g. *(2)*).**
5) **No examination aids, specifically no electronic devices including calculators and electronic dictionaries, are permitted. Use or possession of unauthorized materials will automatically result in the award of zero grade for this examination (FHB 5.1.2.A).**
6) **A minimum of 40 percent must be obtained on this final examination in order to achieve a passing grade in the course.**

Answer all *four* questions in the examination books provided.

[10] 1. a) What is the major difference between AWT and Swing components *(1)*?
b) Explain the AWT event model, i.e., explain what happens when a physical input event such as a key stroke or a mouse movement occurs *(2)*.
c) Explain the `paint-repaint` mechanism of a container *(2)*. What is the advantage of using this mechanism *(1)*, and when should you use it *(1)*?
d) What is meant by active rendering *(1)*? What is the advantage of this approach *(1)*, and when should you use it *(1)*?

[10] 2. Explain two different ways of creating a thread in Java *(2)*. What is meant by a lock of an object and by method synchronization in Java *(2)*? How are locks affected by the methods `wait()` and `notify()` *(2)*? Explain how to use the methods `wait()` and `notify()` to communicate/cooperate among threads (you may use an example) *(2)*. How and when do you use the method `join()` *(2)*?

[10] 3. Choose *one* of the following design patterns Strategy, Singleton or Command. For the chosen pattern, describe the problem the pattern addresses *(2)*, give an example of the use of the pattern for a problem (you may include a class diagram) *(4)* and describe how the pattern solves the problem (you may use a class diagram of the model) *(4)*.

[10] 4. Compare the classes and the functionality provided by the Java API NIO and the combination of the packages `java.io` and `java.net` *(4)*. Explain how to establish a connection using the latter approach *(4)*. Explain how to use a `Selektor` object in the NIO approach *(2)*.