1. **Course Number and Course Title:**
   CMP 256 – GUI Design and Programming

2. **Credits Hours:**
   3 – 2 – 3

3. **Prerequisites and/or Corequisites:**
   Prerequisite: CMP 220 Programming II
   Co-requisites: CMP 256L

4. **Name and Contact Information of Instructor:**
   Dr. Michel Pasquier

5. **Course Description (Catalog Description):**
   Covers the design and implementation of interactive graphical user interfaces (GUI). Provides an introduction to object-oriented Java programming and software patterns, including key GUI-centric features such as Java interfaces, multi-threading, exception handling, and the strategy and model-view-controller patterns. Covers basic 2D graphics operations, essential GUI components, their layout and related interface design principles, and their interactions using the event-driven programming paradigm.

6. **Textbook and other Supplemental Material:**
   Textbook:
   - C.S. Horstmann and G. Cornell, *Core Java,* Vol. I *Fundamentals,* 11th ed*.,* Prentice Hall, 2018.

   Other supplemental material:
   - D. Eck, *Introduction to Programming Using Java,* 8th ed., 2019. Online at: http://math.hws.edu/javanotes/
   - Java Tutorials: *Graphical User Interfaces with Swing,* Oracle, 2015. Online at: https://docs.oracle.com/javase/tutorial/uiswing/
   - Selected program examples, online at: https://ilearn.aus.edu/

7. **Course Learning Outcomes:**
   Upon completion of the course, students will be able to:
   1. Demonstrate an understanding of the implementation principles of modern, interactive graphical user interfaces (GUI).
   2. Create object-oriented programs and GUI-based applications using Java.
   3. Employ 2D graphics primitives to draw text, objects, and images.
   4. Apply software patterns that are central to GUI design, such as the strategy and model-view-controller patterns.
   5. Implement the event-driven programming paradigm via Java interfaces.
   6. Identify the key components of a GUI and manage their placement with the help of layout managers.
   7. Build responsive, multi-threaded GUI applications, such as for animation.
   8. Use state-of-the-art GUI development tools, such as Netbeans and Java Swing.

8. **Teaching and Learning Methodologies:**

Methods include lectures, problem-based learning, class discussions, and group work, as well as laboratory sessions. Students learning is assessed via in-class quizzes, exams, homework, and programming assignments/projects.

9. Course Topics and Schedule:

Topic/Activity	Weeks
Introduction to Java programming	Week #1
Java strings, Unicode, arrays, file I/O	Week #2
Java classes and objects, OO design, exceptions	Week #3
Java inheritance, polymorphism, abstract classes	Week #4
Java callbacks, interfaces, strategy pattern	Week #5
GUI basics and 2D drawing, text, shapes, colors	Week #6
GUI basics and 2D drawing, geometric transforms – Midterm I	Week #7
Multi-threading basics, concurrency, thread synchronization	Week #8
Multi-threading and animation, deadlocks, wait/notify	Week #9
Event-handling, listeners/adapters, GUI events, actions	Week #10
Model-View-Controller pattern, layout managers	Week #11
Swing components and GUI design	Week #12
Swing components and GUI design – Midterm II	Week #13
Swing components and GUI design	Week #14
Revision	Week #15
Final Exam	Week #16