

1. Course Number and Course Title:

CMP256 – GUI Design and Programming

2. Credits Hours:

3 – 2 – 3

3. Prerequisites and/or Co-Requisites:

CMP220 Programming II

4. Name and Contact Information of Instructor:

Dr. Michel Pasquier

Office: EB2-213

Email: mpasquier@aus.edu

Phone: (06) 515-2883

Office Hours: as posted on *iLearn* or by appointment

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*, 10th ed., Prentice Hall, 2015. <http://www.horstmann.com/corejava.html>

Other supplemental material:

- D. Eck, *Introduction to Programming Using Java*, 7th ed., 2014. Online at: <http://math.hws.edu/javanotes/>
- Java Tutorials: *Graphical User Interfaces with Swing*, Oracle, 2012. Online at: <https://docs.oracle.com/javase/tutorial/uiswing/>

7. Learning Outcomes:

Upon completion of the course, students will be able to:

1. Understand the key design and implementation principles of modern, interactive graphical user interfaces (GUI).
2. Create object-oriented programs and GUI applications using Java.
3. Employ 2D graphics primitives to draw text, objects, and images.
4. Recognize and use software patterns that are central to GUI design, such as the strategy and model-view-controller patterns.
5. Apply 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 Java Swing.

8. Teaching and Learning Methodologies:

Methods include lectures, problem and laboratory based learning methods (homework assignments, in-class quizzes, exams, and programming assignments), and class discussions.

9. Course Topics and Schedule:

Topic	Weeks
Introduction to Java programming	2
Java classes, inheritance, exceptions	2
Java interfaces and strategy pattern	1
GUI and 2D drawing basics	2
Multi-threading and animation	1
Event-handling and listeners	1
MVC pattern and layout managers	1
Swing components and design	4
Review and evaluation	2
Total:	16

10. Schedule of Laboratory and other Non-Lecture Sessions:

Assignment	Due Date (tentative)
Lab 1 – Java basics and Netbeans	Week #2
Lab 2 – Java strings, arrays, file I/O	Week #3
Lab 3 – Java classes and objects, exceptions	Week #4
Lab 4 – Inheritance and polymorphism	Week #5
Lab 5 – Java interfaces and strategy pattern	Week #6
Lab 6 – GUI basics, text and textures	Week #7
Lab 7 – Multi-threading and animation	Week #8
Lab 8 – Advanced 2D drawing, transforms	Week #9
Lab 9 – Events, listeners and adapters	Week #10
Lab 10 – MVC and layout managers	Week #11
Lab 11 – Swing components, text input	Week #12
Lab 12 – Buttons, list boxes, sliders, menus	Week #13
Lab 13 – Dialogs and panes, data exchange	Week #14

11. Out-of-Class Assignments with Due Dates:

Assignment	Due Date (tentative)
Homework 1 – Java basics	Week #2
Homework 2 – Classes and inheritance	Week #5
Homework 3 – Interfaces and GUI basics	Week #8
Homework 4 – Events and Swing layout	Week #11
Mini-Project – Swing-based Interactive Game	Week #14

12. Student Evaluation:

Assessment	Weight	Due Date (tentative)
Homework and Mini-Project	15 %	cf. section 11
Quizzes	8 %	Bi-weekly
Lab Assignments	12 %	cf. section 10
Midterm Exam I	15 %	Week #6
Midterm Exam II	20 %	Week #12
Final Exam	30 %	Week #17

13. Contribution of Course to Program Outcomes

This course contributes to the accomplishment of the following CMP program outcomes:

CMP program outcome	Emphasis in this course
a) An ability to apply knowledge of computing and mathematics appropriate to the discipline	
b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution	●
c) An ability to design, implement, and evaluate a computer-based system, process, component or program to meet desired needs	●
d) An ability to function effectively on teams to accomplish a common goal	
e) An understanding of professional, ethical, legal, security and social issues and responsibilities	
f) An ability to communicate effectively with a range of audiences	
g) An ability to analyze the local and global impact of computing on individuals, organizations and society	
h) Recognition of the need for and an ability to engage in continuing professional development	
i) An ability to use current techniques, skills, and tools necessary for computing practice	●
j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modelling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices	

American University of Sharjah | College of Engineering

k) An ability to apply design and development principles in the construction of software systems of varying complexity	●
--	---

Emphasis: ● High; ◐ Medium; ○ Low; Blank – Nothing Specific Expected