

Final Exam is:

Section01: **Wednesday 12/13/2017 8am – 9:50am**

Section02: **Monday 12/11/2017 noon – 1:50pm**

Exam Structure:

- Very similar in design to our quiz questions:
 - Short-answer (~1 paragraph or a drawing or a code-snippet)
 - Code-analysis (I give you the code, you describe what it does)
 - Code-writing (I tell you the desired output, you write the code)
 - Finding / Fixing Errors
- It will take the entire exam time.
- You can have your test next semester, but I'll post your final exam (and class) grade on blackboard.
- Get lots of sleep, eat a good breakfast and start studying early!
- Read through the readings (a couple of times) if you haven't done so already – it helps!
- Look through your old quizzes (if you want a blank copy, email me)
- Look through the slides
- Look through the example programs (and my lab solutions)
- **You are responsible for everything we've covered this semester.** The list below is just for your convenience...
- Good luck☺

Final Exam Topics:

- **Section01: High-Level languages + Python intro**
 - Compilers and Interpreters relation to a high-level language
 - Script mode vs Interactive mode (in IDLE)
 - print statements: expressions, multiple items, sep + end
 - Comments
 - input function
- **Section02: Variables and Types**
 - Statements and Expressions
 - Building blocks of expressions (constants (all types), function calls, ...)
 - Operators (and their precedence and return type)
 - Variables
 - Basic functions (print, input, round, int, str, float)
- **Section03: importing**
 - import modules
 - random module, and its important functions
 - time module, and its important functions
 - pygame (some of this was covered in later sections)
 - Window creation
 - Surfaces
 - Drawing commands
 - Game / Animation loops
 - Making an object move / rotate / update at a constant rate of machines of different speed.
 - Event handling (event-handling and device-polling)

- clock objects
 - font objects
 - blitting and off-screen surfaces (including blitting a portion of an image)
 - transforming (and centering) surfaces
- **Section04: Control Structures**
 - if statements (and its various forms)
 - Boolean constants and expressions (conditions)
 - while statement
 - break and continue statements within
 - Nested if's and while's
- **Section05: Game Loops**
 - The major parts (update, event-handling, drawing)
 - clock objects in pygame
 - frame-rate dependent movement / rotation / etc.
 - event-handling in pygame
 - device-polling (mouse and keyboard)
 - event-handling (mouse, keyboard, resize, quit, etc.)
- **Section06: Sequences**
 - The four types: tuples, strings, lists, dictionaries
 - Operators and functions common to all (e.g. len, +, slicing, indexing, in operator, etc)
 - for loops
 - range objects
 - Nested tuples, lists (aka 2d lists / tuples)
 - dictionaries
 - tuple/list unpacking
 - list methods (append, insert, remove) + del operator
 - mutability vs. immutability
- **Section07: Functions**
 - Goals (abstraction, encapsulation, removing redundancy, decomposition)
 - Parameters and Arguments
 - Default parameters
 - Positional arguments
 - Return values
 - Function invocation steps (parameter-value copying, execution, parameter/local-destruction, return)
 - Mutable objects and parameters
 - Scope (global vs. local variables)
 - Making your own module.
- **Section08: OOP**
 - Goals (abstraction, encapsulation)
 - Basic syntax
 - methods
 - docstrings
 - Relationship between classes and objects.
 - Attributes and methods
 - The role of self
 - `__init__` and `__str__` methods.
- **Section09: Trig**
 - Angles / Orientation

- Polar \Leftrightarrow Cartesian coordinates
- Vector motion
- "Rotate-able" points (e.g. the turret-points on the ship in Lab8)
- Pointing towards something (atan2)
- Sin & Cos for other purposes (e.g. fading color)