

Outline:

For this final culminating activity, you are to **individually** create a program/game of your choice using one of the coding tools we covered this semester, (Scratch; Python; HTML). **The assignment, including all required parts, MUST be submitted no later than Thursday, January 26, 2017.**

Examples include:

- Calculator (Python, Scratch)
- Magic Eight Ball (Python)
- Card Game / Poker Hand Game (Python, Scratch)
- Slot Machine (Python, Scratch)
- Choose-Your-Own Adventure Story (Scratch, HTML)
- Trivia Quiz (Scratch, HTML)
- Text-Based/Graphic-Based Adventure Game (Python/Scratch/HTML)

Requirements:

1. Software Development Cycle

a. Project Outline

Complete the **Project Outline Handout**, which outlines the type of program you are creating, with a brief description/set of instructions/purpose/etc. You must also identify the coding concepts, (e.g. loops, libraries, dictionaries, logic, etc.) that you will use as well as how they are used in your program.

DUE: Monday, January 9, 2017

b. Daily Work Log

You must take a few minutes at the end of each period to outline the work you have completed during class time. It is important to be accountable for your time. You may also indicate work you've done outside of class.

This must be saved on Shares/From Students to Staff/Wilson/ICS20/Final Project at all times so that I can access the file.

2. Interface Layout

- a. Before programming, think of an outline of what each screen will look like when finished (i.e. intro, main menu, credit screen, etc...
- b. Include an "**About**" screen with information about the program and programmer (i.e. your name, date, and class code (ICS20)).
- c. Include a "**Help**" screen with the basic rules, how to play, and/or use the application.
- d. User friendly layout.
- e. Use, colour, pictures, font changes to spruce up the appearance of the application.
- f. Experiment with the image and graphics tools (not applicable for Python).

3. Write the Code:

a. Declare all variables with appropriate names.

This will depend on the programming language you use, but your code must be clear and easy to follow. You should also include appropriate comments in your code that include your name, the date of creation, the course code, etc.

b. Create a fully functioning program.

As a culminating project worth 30%, it is important that you create a program that shows what you have learned this semester. A very basic, short, simple program will not cut it. You must take the time to plan an appropriate program, design it, code it, test it, and share it. It must also be **original**.

c. Test and validate your program.

You should make use of a partner for this step. What does your tester like/dislike about your program? What could be improved? Take time to collaborate.

d. Share it.

Ensure that a completed copy of your program and an original copy of your code are made available. If you are using *Scratch*, this is pretty easy. If you're using something else, it may involve a bit of extra work.

4. Marks and Success Criteria:

In addition to marks, during the project, each day I will be using the following criteria to evaluate your learning skills:

- Time management
- Use and understanding of content previously taught
- Self-learning (review/research stuff!)
- Reliance on teacher and others – Can you figure stuff out on your own, (or by going online)
- Daily reports
- Classroom etiquette
- Shows knowledge of a specific coding application/tool
- Shows ability to research appropriate tools/concepts
- Shows ability to problem-solve effectively

In order to assist you, you should use the six steps of the design process when planning. You are responsible for being able to discuss all of these points when conferencing with me throughout your project.

Analysis:

- Define the problem/purpose of what your software will accomplish
- State the requirements and expectations
- Create a reasonable timeline for your projected progress

Design:

- Explain how the program will work
- Provide step by step details of your plan including diagrams (storyboards, flow charts, pseudo code, etc)

Programming:

- Create the program making sure to incorporate the concepts taught in class (procedures, arrays, loops, etc)
- Note; you may need to do your own research on topics not yet covered in class however you can create a great project using all that has been taught.

Documenting:

- Create both internal AND external documentation
- Provide extensive detail explaining the use of your program
- All code should be organized using comment separators
- All code should be explained using comments

Testing:

- Your program should be error proof
- Create a report of the errors/problems encountered and how you modified your program to accommodate to them

Maintaining:

- Look back at your initial plan. Did you complete the expectations?
- Have others run your program and make appropriate changes based on their recommendations

Name:

ICS20 Course Culminating Rubric

Date:

	Level 4	Level 3	Level 2	Level 1	Level R
Planning Program was planned out well. All required checkpoints were met as expected. Program was developed with little to no help.	Student clearly expressed their goals, followed a plan closely, and independently resolved issues along the way with minimal assistance. All deadlines were met.	Student had a fairly clear picture of goals but struggled to implement their plan. Most deadlines were met.	Student brain stormed but struggled to implement their plan without considerable teacher guidance. Only some deadlines were met.	Student spent little time brainstorming and was often unclear on how to continue without constant teacher assistance. Most deadlines weren't met.	No evidence that planning was undertaken.
Code Organization Clear internal documentation used including header, variable details, purpose of loops and conditional statements. Organized into appropriate sections with comment separators. External documentation describes the functions of the program and user input expectations.	Code is easy to follow with descriptive variables, extensive commenting and well organized. External documentation provides extensive details on usage and requirements of program.	Good degree of commenting and organization used to explain essential portions of the program. External documentation explains usage of program clearly.	Some commenting used to explain some code sections. Code is understandable but not well organized. External documentation explains some parts of the program.	Code is hard to follow with limited use of commenting to explain code. Slapped together with no organization. External documentation does not fully explain program.	No evidence that coding was completed.
Code Content Proper use of coding structure including input/output statements, variables, loops and if-then statements. Code is logical and well thought out.	Code is well thought out, and logical. All components are used appropriately.	Mostly well thought out logic, but not optimized. Most components used correctly.	Some thought put into program but doesn't flow properly. Many components are used incorrectly.	Code is full of programming errors with few components used properly.	No evidence that coding was completed.
Testing In-house (your own), 1 internal (classmate), and 2 external (friends/family) tests sessions completed. Reports submitted stating issues discovered and steps taken to resolve any problems.	All testing completed and fully documented.	Some testing completed with basic documentations.	Some testing complete with minimal documentation.	No testing done beyond typical daily error resolution.	No evidence that testing was done.
Code Functionality Program compiles and executes as expected. The interface is easy to follow and user friendly. The final product functions as expected and as according to the initial plan.	No flaws, functions as expected, and very user friendly.	Some minor flaws, but still quite functional and easy to use.	Some major flaws, but code still executes and some parts work.	Major flaws, code won't execute, but is present.	No evidence that coding was completed.

Comments: