



## Create Project – Applications from Ideas<sup>1</sup>

### 70 points

### Overview

Programming is a collaborative and creative process that brings ideas to life through the development of software. Programs can help solve problems, enable innovations, or express personal interests. In this project, you will design and implement a program to address an issue that is important to society or to you personally. Developing a program of your choice will help to deepen your understanding of the programming process and the role it plays in computing. This project is open-ended, with a few technical requirements (see below). What project to choose? Maybe something that'll help you in your major? Maybe something that you have passion about? Maybe something that'll improve your daily life (e.g., an exercise tracker with statistics)?

### General Requirements

In this project, you will plan, design, implement, and test your program using the Python programming language. You will provide written responses to prompts about your program and specific code segments that are significant to the functionality of your program. It is recommended that a portion of the program development involve some form of collaboration with another student in your class, for example, in the planning, designing, implementing, or testing (debugging) part of the development process. The code submitted needs to be written by you (i.e., this is not a group project), but you should ask your peers for help on the **creative aspects** of the project (e.g., "I'm doing an exercise tracker and have added 'total miles run per week' feature; what other features should I add?") or ask if you are stuck on what project to create (e.g., "any thoughts on what I could do?").

You are required to:

- develop a program in Python that processes input, produces output, and uses (1) at least one loop, (2) at least one function you have defined, (3) at least one list, and (4) file input/output (or graphics);
- Collect creative feedback from your peers during the design phase
- develop test cases to be used to determine whether your program performs correctly;
- write responses to all prompts below; and
- submit your entire program code.

### Deliverables

#### 1. Written Response #1 (max 300 words): 8 points DUE: October 26<sup>th</sup> at 11:45pm

Submit a nicely formatted pdf document to Gradescope that includes clearly labeled responses to all prompts. You do not need to do any coding to write this response. The purpose is to get you thinking about what you will create. **Your response to all prompts combined must not exceed 300 words (and needs to be at least 200 words); place total # of words for 1a-1d on bottom of page. Place your answers in order (1a – 1d) with the bolded titles listed below:**

---

<sup>1</sup> This project is modified from the AP Computer Science Principles course.



1a. **TITLE:** Include your name, your CSCI 101 section, and a project title that accurately describes what you are planning to create.

1b. **PROGRAM:** Describe the program you will write and why you decided to write this program. Identify the input to your program, the output that your program will generate, and how the input will be used to create the output.

1c. **PROCESS:** Explain how you plan to develop the program and what form of collaboration you will use in the process. Be specific and identify who you will be collaborating with and the role of each member of the collaboration.

1d. **CONCERNS:** Do you have any special needs or concerns in the development of this project? Will you need extra help on anything to meet the goals of your chosen project?

NOTE: The CS101 team will read your Written Response #1 ASAP and will email you if the project you've proposed seems too complex for the time you have to create it.

## 2. **Written Response #2 (max 500 words): 17 points DUE: November 9<sup>th</sup> at 11:45pm**

Submit one nicely formatted PDF document to Gradescope that includes clearly labeled responses to prompts 2a and 2b below; please also copy your prompts from 1a and 1b as a reminder on what you are doing. **Your response to the new prompts (i.e., 2a and 2b, not including code) must not exceed 500 words (no minimum required); place total # of words for 2a-2b on bottom of page. Place your answers in order (2a then 2b) with bolded titles listed below.**

1a and 1b. Copy your previous prompts and place at the top of this pdf document.

2a. **COLLECTION:** Copy and paste a program code segment showing the file I/O (or graphics) used in your program. Describe what the data represents in your program and how it'll be used.

2b. **ALGORITHM:** Copy and paste a function from your program. This function must use at least one parameter.

- Explain how the function contributes to the overall functionality of the program.
- Describe what the advantages are for creating this function.
- Explain how the selected function works.
- Identify two test cases that were used as input to the function. Each test case should demonstrate a different condition to help determine whether the function works properly.
- Describe what conditions are being tested by each test case and the expected result of using each test cases.

## 3. **Written Response #3 (max 200 words): 10 points DUE: December 7<sup>th</sup> at 11:45pm**

Submit one nicely formatted PDF document to Canvas that includes clearly labeled responses to prompts 3a and 3b below; again, copy your prompts from 1a and 1b as a reminder on what you are doing. **Your response to the new prompts (i.e., 3a and 3b) must be between 150 and 200 words; place total # of words for 3a-3b on bottom of page. Place your answers in order (3a then 3b) with bolded titles listed below.**



1a and 1b. Copy your previous prompts and place at the top of this pdf document.

3a. **EXECUTION:** Provide documentation on how to execute your program. Include at least one example that illustrates the execution; do NOT count toward word count. Provide details on anything the grader needs to know to execute your program successfully. If it would help the grader, be sure to submit an example input file to Canvas as well (zip input file(s) AND code together into one file).

3b. **REFLECTIONS:** Provide a paragraph on what you learned from this project. It might help to think about what problems you encountered, and what you would do differently (if you had to do it again).

#### 4. Program Code: 35 points DUE: December 7<sup>th</sup> at 11:45pm

Submit the Python file that contains your code to Canvas. The file you submit must execute in Python 3. If your program requires an input file of a certain type/format to execute, upload a zip file that contains your Python code and example input file(s) in one zip file.

### To do well on this project:

#### You must:

- Write a significant program. This is hard to quantify, but it should be something that you would want to show to people important in your life;
- The function you select in 2b cannot be trivial; it must play a significant role in your program;
- Allow your interests to drive the development of your program (what do you want to create? what would be useful to you?); and
- Do NOT procrastinate. (Students working on final projects for CSCI courses have previously said “gosh, wish I had started earlier”. No one ever says “gosh, wish I had started later.”)

#### You may:

- Seek assistance from any instructor, TA, or mentor in defining your focus and choice of program;
- Seek assistance from any instructor, TA, or mentor to resolve technical problems that impede work; and
- Collaborate with other students during the design process and while testing and debugging your program.

#### You may not:

- Submit code that has been revised, amended, or edited by another individual. In other words, you must write your own code; and
- Seek assistance or feedback on answers to the prompts.

#### Other thoughts from former CSCI 101 students:

- Debugging was a larger part of this project than expected. Be sure to give yourself ample time to debug.
- Functions are extremely important for large projects. Use functions liberally to keep your code clear.
- Libraries can be difficult to learn but are very useful for quickly coding really powerful applications.

### Extra credit opportunity (10 points):

Will you create something that will “wow” your peers? If yes, you may be chosen to present your project in class and earn 10 points extra credit. To be considered for this extra credit opportunity, you need to get your project done EARLY (i.e., the final deliverables, #3 and #4 above, need to be submitted to Canvas four days early, by December 3<sup>rd</sup>, 11:45pm). Top projects will be presented in class on December 6<sup>th</sup>.