

Overview: Beginning with Stepping Stone Two in Module Three, you will complete a series of related stepping stone labs that will help you build object-oriented programming skills that relate to your final project. Be sure to incorporate feedback from your instructor as you develop your work through the stepping stone labs and develop your final project.

Note that Stepping Stone Lab One: Pseudocode is graded with a separate rubric.

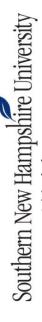
Stepping Stone Labs Two through Six specifics:

Stepping Stone	Module	Title/Topic	Task Specifics
Two	Three	Data Types	Produce basic data types for a recipe manager program.
Three	Four	Validating Input With Branches	Develop a branched structure for a recipe manager program.
Four	Five	Entering Ingredients With Loops	Develop iterative loops for a recipe manager program.
Five	Six	Collection and Item Class With Accessors and Mutators And create a test class.	Develop the first version of the Recipe class and create a test class.
Six	Eight	The RecipeBox Driver Application	Produce a driver application for a recipe manager program.

Each of your programming assignments includes two parts:

- Code file(s)
- Written reflection as text file or Word document

The assignment parts are submitted together with the written submission and with the code as a file attachment.



The following critical elements should be addressed in your project submission:

Code Reflection

A brief explanation of the code, its purpose, and a brief discussion of your experience in developing it, including any issues that you encountered while completing the stepping stone and what approaches you took to solve them

II. Specifications

Source code must meet its specifications as defined by the data and problem. However, this may require multiple attempts or iterations. You will be given credit for code that is well on its way to meeting specifications or solving the problem.

III. Correctness

will receive credit for producing fully functioning code (producing no errors) that aligns with as many of the specifications as possible. Note: You should Source code must behave as desired. While correct code produces the correct output as defined by the data and problem, for the stepping stones you write your code in such a way that the submitted files execute, even if they do not produce the correct output.

IV. Readability

Code needs to be readable to a knowledgeable programmer. In this course, readable code requires the following:

- Consistent, appropriate white space (blank lines, spaces) and indentation to separate, distinct parts of the code and operations
- Explicit, consistent variable names that clearly indicate the data they hold and are formatted consistently
- Organized structure—clear design separating components with different responsibilities

V. Annotation

All code should also be well-commented. This is a practiced "art" that requires striking a balance between commenting everything, which adds a great deal of unneeded noise to the code, and commenting nothing. Well-annotated code requires you to do the following:

- Explain the purpose of lines or sections of your code, detailing the approach and method you took to achieve a specific task in the code.
- Document any section of code that is producing errors or incorrect results.

Ruhric

Your assignment should be submitted as a zip file of the exported code and reflection text, as required.

Note that, although the stepping stone labs are graded, their main purpose is to provide useful practice and feedback that you can incorporate as you build the knowledge and skills you need to succeed in the final project.

Southern New Hampshire University

Critical Element	Proficient (100%)	Needs Improvement (70%)	Not Evident (0%)	Value
Code Reflection	Describes purpose of the code, techniques	Lacks details of code purpose, techniques	Does not describe purpose of code, techniques	20
	implemented to solve the problem, challenges	implemented, or challenges encountered	used, or challenges encountered	
	encountered, and the approaches to			
	overcome the challenges			
Code	All or most algorithm specifications are fully	Details of the specifications are not met in	The program does not meet the specifications	20
Requirements	met	significant instances		
Code Correctness	The program functions as designed in most	The program functions as designed in limited	The program does not function as designed	20
	cases	cases		
Code Readability	Code follows proper syntax and demonstrates	Code contains variations from established	Code exhibits consistent and significant	20
	deliberate attention to spacing, white space,	syntax and conventions	variations from established syntax and	
	and variable naming		conventions	
Annotation	Code annotations explain and facilitate	Code annotations are incomplete or provide	Code annotations do not explain the code, do	20
	navigation of the code	insufficient assistance with understanding the	not facilitate navigation of the code, or are not	
		code	present	
			Total	100%