Assessment 2: Plan and Implement a Terminal Application

Overview

Assessment Title	**Plan and Implement a Terminal Application**
Release Date	Term 1, Week 5, Monday
Due Date	Term 1, Week 6, Monday
Subject	Foundation Programming - CMP1041 Programming I - PRG1002
Weighting	71% of final mark for Foundation Programming 45% of final mark for Programming I
Course	Diploma of Information Technology - Bootcamp Delivery Mode (FlexTrack)
Assessment Policies	Coder Academy utilises the Academy of Information Technology's policies relating to assessment, academic matters and student welfare. If you have any questions about these policies, please speak to your educator, or Coder Academy's student services team. A penalty of 10% of the total mark per day will apply for late submissions, inclusive of weekends.
Learning Outcomes	**Foundation Programming** CMP1041-1 Demonstrate an understanding of programming concepts CMP1041-2 Plan and create functional programmes CMP1041-3 Design and outline a software development plan CMP1041-4 Successfully conclude a project according to design specifications and testing CMP1041-6 Demonstrate practical knowledge of the use of software development tools **Programming I** PRG1002-3 Demonstrate an understanding of the underlying principles of programming language PRG1002-4 Employ different techniques pertaining problem solving skills PRG1002-5 Demonstrate algorithmic thinking and apply it to programming
Result	90 marks for Foundation Programming 40.5 marks for Programming I

Context

Terminal applications, also known as command line applications, are run from the shell and accept input from stdin and arguments such as flags and options, and produce an output that is useful to the user.

You will create your own terminal app because:

- terminal applications are an essential part of learning to utilise the functions of all operating systems efficiently and in a way that promotes your understanding of the inner workings and processes of operating systems and the hardware itself;
- many applications used across IT functions utilise terminal applications as a standard, rather than applications with a graphical user interface;
- developing your own terminal application will increase your technical sophistication and ability to use other terminal applications.

Project Brief

You are to design, implement and test a terminal application and throughout the process demonstrate that you are able to use a range of developer tools.

The mandatory requirements/constraints for your application are:

- it must utilise arguments that can be provided in the terminal when running the application (args, options and flags)
- accept user input in the form of a file or text input
- produce printed output or interact with the file system

You must consult with your educator on the application you intend to make before completing substantial work on the ***software development plan*** *or implementing the application itself.*

Assessment Requirements

Code	Requirement	Submission Requirements	pg1
	Design a software development plan for a terminal application. **NOTE:** we do not accept submission of Word, Pages or Open office documents. We will only accept PDFs, or markdown files for written assessments.	This task should be written in a single document in a report format with headings. It is suggested that this document is placed in the root of your source control repository, and that you write it in markdown.	
	Statement of Purpose and Scope		
T1A2-1	**Develop** a statement of **purpose** and **scope** for your application. It must include: - **describe** at a high level what the application will do - **identify** the problem it will solve and **explain** why you are developing it - **identify** the target audience - **explain** how a member of the target audience will use it	300 - 500 words. Please include this in your **software development plan** as a separate section with an appropriate heading.	CMP1041-3.1
	Features		
T1A2-2	**Develop** a list of features that will be included in the application. It must include: - at least THREE **features** - **describe** each feature **Ensure** that your features include the following language elements and concepts: - use of variables and the concept of variable scope - loops and conditional control structures - error handling **Note:** If the features you described do not require you to use all of the above it is likely that your application is not sophisticated enough. **Consult with your educator** to check your features are sufficient to address the criteria for T1A2-7.	300 words (approx. 100 words per feature) Please include this in your **software development plan** as a separate section with an appropriate heading.	CMP1041-3.1

	User Interaction and Experience		**pg2**
T1A2-3	**Develop** an **outline** of the user interaction and experience for the application. Your outline must include: - how the user will find out how to interact with / use each feature - how the user will interact with / use each feature - how errors will be handled by the application and displayed to the user	Please include this in your **software development plan** as a separate section with an appropriate heading.	CMP1041-2.1, CMP1041-2.3
	Control Flow Diagram		
T1A2-4	**Develop** a diagram which describes the control flow of your application. Your diagram must: - show the workflow/logic and/or integration of the features in your application for each feature. - utilise a recognised format or set of conventions for a control flow diagram, such as UML.	Please provide a image file or PDF of your diagram. Please include this in your **software development plan** as a separate section with an appropriate heading.	CMP1041-3.2
	Implementation Plan		
T1A2-5	**Develop** an implementation plan which: - outlines how each feature will be implemented and a checklist of tasks for each feature - prioritise the implementation of different features, or checklist items within a feature - provide a deadline, duration or other time indicator for each feature or checklist/checklist-item	You must submit this as a written document, ideally in a tabular format, and it is suggested that you enter your checklists into an appropriate project management application to assist you in completing T1A2-10. Your checklists for each feature should have at least 5 items.	CMP1041-3.3
	Status Update		
T1A2-6	Write **TWO** status updates on your application. You must write a status update when: - you significantly change a feature or your implementation plan at any point - encounter a significant issue, challenge or roadblock - are halfway through implementing the features - prior to / after testing the application	You should submit these logs throughout the course of developing your application. Each status report should be between 100 - 200 words. The **Development Log** should be written in markdown and placed in a file named development-log.md in the root of your source control repository. Ensure you put the date of the log at the top of each log entry. Please send your log to your educator as a markdown file as you complete them. This is so we can provide you with feedback.	CMP1041-2.2

	Implement Application		**pg3**
T1A2-7	**Implement** all features in the software development plan you have designed. You must utilise a range of programming concepts and structures using Ruby such as: - variables and variable scope - loops and conditional control structures - write and utilise simple functions - error handling - input and output - command line arguments - importing a Ruby Gem - using functions from a Ruby Gem		CMP1041-1.1, CMP1041-1.2, CMP1041-2.3, PRG1002-5.1, PRG1002-5.2, PRG1002-5.3, PRG1002-3.1, PRG1002-3.2
T1A2-8	**Apply** DRY (Don’t Repeat Yourself) coding principles to all code produced.		PRG1002-3.3
T1A2-9	**Apply** all style and conventions for the programming language consistently to all code produced.		CMP1041-1.3
T1A2-10	**Creates** an application which runs without error and has features that are consistent with the development plan.		CMP1041-4.3
T1A2-11	**Design** a help file which includes a set of instructions which accurately **describe** how to use and install the application. You must include: - steps to install the application - any dependencies required by the application to operate - any system/hardware requirements - a written explanation of the different features of the application	The help file should be 100 - 200 words.	CMP1041-4.2
	Test Application		
T1A2-12	**Design** TWO tests which check that the application is running as expected. Each test should: - cover a different feature of the application - state what is being tested - provide at least TWO test cases and the expected results for each test case	An **outline** of the testing procedure and cases should be included with the source code of the application and written in markdown.	CMP1041-4.1

	Developer Operations		**pg4**
T1A2-13	**Utilise** source control throughout the development of the application by: - making regular commits (at least 20 commits) with a commit message that summarises the changes - pushing all commits to a remote repository	You must submit a source control repository as a zip file to satisfy this requirement.	CMP1041-6.1
T1A2-14	**Utilise** a project management platform to track the development of the application by: - having features itemised and broken down into checklists - setting deadlines, duration or a time-frame for each feature or task - prioritising tasks	You must submit screenshots, or a valid export from a project management application to satisfy this requirement. If you are using a non-standard project management application, please discuss with your trainer whether they can access the file. If in doubt, screen capture your project management application and submit screen captures.	CMP1041-6.2
T1A2-15	**Utilise** developer tools to automate the building and testing of the application by: - writing a script which runs the tests - writing a script which turns the application into an executable; OR - packaging the application for use as a module or dependency	You must submit a script file in with your source control repository to satisfy this requirement.	CMP1041-6.3
	Presentation		
T1A2-16	Present your terminal application to the class. You must provide a walk-through of the logic of your application and how the application is used. The time limit for your presentation is 5 minutes.		PRG1002-4.4

Glossary

Describe means to provide detail.

Identify means to make a statement or name something.

Explain means to provide detailed reasons/information relating to why or how something works

EXTRA BIT TAKEN FROM SIMO’s SCREENSHOT

Deliverables

Following is a list of deliverables (which should cover all the above requirements) to include in your submission. The table also details how the information should be organised.

Deliverable	Description	Location/Folder (in Zip File)
README.md	General project documentation is to be compiled as a single markdown file named README.md	/ (Root folder of your zip file)
Resources	All files linked by the README.md file must be included in this folder must be in either png, jpeg, pdf, or markdown (md) format	docs/
Presentation	Slide deck of your presentation in pdf format	ppt/
Source Code	Source code for your entire project	src/

Note: All links to online material should have corresponding screenshots included in submission

******* THIS IS NOT A COMPLETE COPY OF THE ASSIGNMENT PAGE********
Pls cross reference to see what is here and what isn’t eg: file format