PENRITH ANGLICAN COLLEGE
To Serve Christ

2025

Assessment Task 1 Notification
Year 11 Software Engineering
Individual Design Project

Task number:	Task 1
Date of Issue:	Tue 25 February 2025
Date & Time Due:	Fri 9 May 2025 (Week 2 Term 2)
Weighting:	30%
Marks:	60
Topic(s) Assessed:	Programming Fundamentals
Outcomes Assessed:	SE-11-01 - describes methods used to plan, develop and engineer software solutions. SE-11-02 - explains how structural elements are used to develop programming code. SE-11-06 - applies tools and resources to design, develop, manage, and evaluate software. SE-11-07 - implements safe and secure programming solutions.
Focus Disposition:	Reasoning Innovating Planning Persisting Focusing Respecting

Rationale of the task
Students explores base concepts of understanding requirements, development and testing stages from the perspective of both a user and a developer. Students also explore the development and use of algorithms, data, data types, data structures and the role design plays in the development of code and the final program.

Feedback provided

- Peer and self-reflection and feedback
- Verbal and written feedback on software solution and security report.

Task description

You are to design, code and document a software solution for an educational product. The software is to be developed in Python.

The educational piece of software is to be an interactive solution. The software could be designed for either primary or secondary students. Students will need to choose a section of the curriculum associated with their chosen year group. For example, the software could help students learn their times table, recognise types of triangles, or learn a new language. It will include an intuitive command line interface that displays directions and responses to the user's requests.

PENRITH ANGLICAN COLLEGE
To Serve Christ

2025

Submission details

The format of the task submission should be one document highlighting documentation and uploading the piece of educational software designed.

• Component A – Projection Documentation
  ○ Requirement definitions
  ○ User specifications
  ○ Pseudocode and flowcharts
  ○ Discussion of data types and data structures
  ○ Data dictionary
  ○ Define and discuss debugging and testing
	30 marks

• Component B – Software Solution A zip file of any implemented solution	30 marks

PENRITH ANGLICAN COLLEGE
To Serve Christ

2025

What I need to do	
Software Development Explore the fundamental development steps in relation to their project.	Write a requirements definition for the given real-world problem. Identify the user specifications for the chosen software solution.
Designing Algorithms	Develop a structured algorithm using pseudocode or flowcharts including the use of subprograms and passing parameters. Algorithms should include sequence, selection, iteration and subprograms, and be described using a structure chart.
Data for Software Engineering	Define and discuss the use of the following data types. Select a minimum of 3 to use in your project: • char and string • Boolean • real • single precision floating point • integer • date and time. Create a data dictionary for use with your project. Define and discuss the use of the following data structures. Select a minimum of 2 to use in your project: • arrays • records • trees • sequential files.

PENRITH ANGLICAN COLLEGE
To Serve Christ

2025

What I need to do	
Developing Solutions with Code	Convert your algorithm into code using: • control structures • data structures • standard modules • subprograms (including parameter passing). Define and discuss the following debugging tools used in your project: • breakpoints • single line stepping • watches • interfaces between functions • debugging output statements. Document and implement at least one appropriate data structure that supports data storage. Describe the errors you experienced in the coding of your solution including: • syntax • logic • runtime.

PENRITH ANGLICAN COLLEGE

To Serve Christ

2025

Name: _____

Class Teacher: _____

Section	Mark	Total
Software Development		**6**
Designing Algorithms		**12**
Data for Software Engineering		**12**
Developing Solutions with Code		**30**
TOTAL		**60**
		%

Penrith Anglican College Years 10-12 Assessment Task Rules

- Submission of assessment tasks is necessary to satisfy course outcome requirements.
- Non-submission of an assessment task will result in a zero mark and an "N" Award Warning letter sent home to inform parents and guardians of the non-submission.
- A late submission of a task in Years 10-12 will result in a zero mark. A letter will be sent home to inform parents and guardians of the late submission.
- Any student who is absent from the College the day before an assessment task is due needs *a medical certificate* (or evidence of other misadventure) explaining the absence and present it to the Head of Department on his/her first day of return to the College. Failure to provide this will result in a 20% penalty being applied to the result in Years 11-12. Any planned, non-medical reason for absence the day before an assessment task is to be approved by the Director of Studies beforehand.

- Any student who is absent from the College without prior arrangement on the day an assessment task is to be completed or handed in:
  - Must submit their task electronically on the due date.
  - Must, if the task requires physical submission, complete *an illness/misadventure form* and attach *a medical certificate* explaining the absence and present it to the Head of Department on his/her first day of return to the College. Students and parents must NOT automatically assume that the reason they have given is acceptable.
  - May, if the task is or involves an in-class component, need to complete the task or a replacement task. The task and the time to complete the task will be at the discretion of the Director of Studies after consultation with the relevant Head of Department.

Please refer to the assessment booklet and academic honesty policy for more details.

PENRITH ANGLICAN COLLEGE

To Serve Christ

2025

Assessment Rubric

Criteria	Limited	Basic	Sound	High	Outstanding
Criteria 1 **Explore the fundamental development steps in relation to their project** **(Mark out of 6)**	Student labels the fundamental development steps in a project. (1)	Student identifies the fundamental development steps in their project. (2)	Student outlines and refers to the fundamental development steps in their project. (3)	Student explains and applies fundamental development steps to their project. (4)	Student comprehensively explains and applies the fundamental development steps to their project. (5-6)
Criteria 2 **Designing algorithms** **(Mark out of 12)**	Student identifies the control structures in an algorithm. (1-2)	Student develops an algorithm which makes some use of the control structures. (3-5)	Student develops a correct algorithm that correctly uses all of the control structures including sequence, selection, repetition and subprograms. (6-8)	Student develops a correct algorithm detailing the steps required for a solution to their problem. The algorithm makes the correct use of all the control structures including sequence, selection, repetition and subprograms. (9-10)	Student designs, develops and desk checks structured algorithms documenting the steps required for a solution to their problem. The algorithm makes the correct use of all the control structures including sequence, selection, repetition and subprograms. (11-12)

PENRITH ANGLICAN COLLEGE

To Serve Christ

2025

Criteria	Limited	Basic	Sound	High	Outstanding
Criteria 3 **Data for software engineering** **(Mark out of 12)**	Student labels data types used in coding. Student identifies a Data Dictionary. Student identifies the data structures used in coding. (1-2)	Student defines the data types used in their project. Student fills in an incomplete Data Dictionary. Student defines the data structures used in coding. (3-5)	Student discusses the data types used in their project. Student creates a Data Dictionary for their project. Student defines the data structures used in their project. (6-8)	Student explains the use of data types used in their project. Student creates a correct Data Dictionary for use with their project. Student explains the data structures and discusses their use in their project. (9-10)	Student justifies their selection of data types used in their project. Student creates a comprehensive Data Dictionary for use with their project. Student justifies the data structures selected for their project. (11-12)
Criteria 4 **Developing solutions with code**	Student completes missing code to solve a task description. Student documents use of data structures and errors.	Student modifies code to solve the task description. The code includes some use of the control structures, some use of data structures and parameter passing.	Student creates code to solve the task description. Their code includes some use of the control structures, some use of data structures and parameter passing	Student creates and documents code to solve the task description. Their code includes correct use of most of the control structures, effective use of data structures and parameter passing.	Student creates, documents and explains efficient code to solve the task description. Their code includes correct use of all the control structures, effective use of data structures and parameter passing.

PENRITH ANGLICAN COLLEGE
To Serve Christ

2025

Criteria	Limited (1-5)	Basic (6-14)	Sound (15-20)	High (21-25)	Outstanding (26-30)
(Mark out of 30)		Student identifies their use of at least one software debugging tool. Student correctly describes at least one error experienced in the coding of their solution	Student defines and documents their use of at least one software debugging tool and describes at least one error experienced in the coding of their solution.	Student defines and documents their use of a range of software debugging tools. Student correctly describes a range of errors experienced in the coding of their solution.	Student correctly uses, defines and documents in detail a range of software debugging tools. Student correctly describes in detail a range of errors experienced in the coding of their solution and how they were corrected.

Comments: