Task Number	1	Task Name	Programming methodology and planning
Course	Software Engineering	Faculty	TAS
Teacher	Ms Moverley	Head Teacher	Ms Godden
Issue date	Tuesday, 2nd April	Due date	Thursday 16th May
Focus (Topic)	Programming Fundamentals	Task Weighting	30%

Outcomes

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
SE-11-08 applies language structures to refine code

Task description

Programming Methodology and Planning
You are to design, code and document a software solution to simulate the dice game of Lucky Sevens. The rules are:
- roll two dice
- if the sum equals 7, win $4
- if the sum is not 7, lose $1

Your program should show the user how much they start the game with and should then:
- simulate the dice roll and show what was rolled
- show whether the user won or lost and what their funds are
- the user can continue to roll until they lose all their money

The program is to be developed in PyCharm and PyGame.

Marking Guidelines

Software Development
- Write a requirements definition (5 marks)
- Identify the user specifications (5 marks)

Designing Algorithms
- Develop a structured algorithm using pseudocode or flowcharts. Your algorithm should include sequence, selection, repetition and subprograms (15 marks)
- Create a structure chart to describe your algorithm (5 marks)

Data for Software Engineering
- Use a minimum of 3 data types in your project from: Char and string, Boolean, Floating point, Integer and use a list data structure (5 marks)
- Create a data dictionary for your project (10 marks)

Developing Solutions with Code
- Convert your algorithm into code using: Control structures, Data structures, Subprograms (50 marks)
- Outline how you used the following debugging tools in your project: Breakpoints, Single line stepping, Watches, Debugging output statements, Peer checking, Desk checks (5 marks)

ASSESSMENT CRITERIA:
Students will be assessed on their ability to explore the fundamental development steps in relation to their project.

CRITERIA	MARK/ GRADE
Outstanding – extensive knowledge, understanding and skills are displayed through a response in which the student comprehensively explains and applies the fundamental development steps to their project	90-100
High – thorough knowledge, understanding and skills are displayed through a response in which the student explains and applies fundamental development steps to their project.	75-89
Sound – satisfactory knowledge, understanding and skills are displayed through a response in which the student outlines and refers to the fundamental development steps in their project.	60-74
Basic – limited knowledge, understanding and skills are displayed through a response in which the student identifies the fundamental development steps in their project.	50-59
Limited – elementary knowledge, understanding and skills are displayed through a response in which the student labels the fundamental development steps in a project.	0-49
Late submission – no misadventure or Assessment not submitted	0 N Warning notification

Assessment Task – Marking Sheet

Component of Task	What you need to submit	Marks
Software Development	Write a requirements definition.	/5
	Identify the user specifications.	/5
Designing Algorithms	Develop a structured algorithm using pseudocode or flowcharts. Your algorithm should include sequence, selection, repetition and subprograms.	/15
	Create a structure chart to describe your algorithm	/5
Data for Software Engineering	Use a minimum of 3 data types in your project from: Char and string, Boolean, Floating point, Integer and use a list data structure	/5
	Create a data dictionary for your project	/10
Developing Solutions with Code	Convert your algorithm into code using: • Control structures • Data structures • Subprograms	/50
	Outline how you used the following debugging tools in your project: • Breakpoints • Single line stepping • Watches • Debugging output statements • Peer checking • Desk checks	/5
	TOTAL	/100

Feedback: