



JOHN EDMONDSON HIGH SCHOOL

Assessment Notification

Faculty: Industrial Arts Course IST Year: 10

Assessment Task: Creating python game

Assessment Weighting: 20% Due: Term 3 Week 4 Date: 16/08/2024

Task Type: Hand in Task In Class Task Practical Task

Outcomes assessed (NESA)

- 5.2.1 describes and applies problem-solving processes when creating solutions
- 5.2.2 designs, produces and evaluates appropriate solutions to a range of challenging problems
- 5.2.3 critically analyses decision-making processes in a range of information and software solutions

Task Description/Overview

Creating a guessing number game with help of python programming language.

Detailed Assessment Task Description

Task Overview: Python Programming

You are to develop a program that will play a simple guessing game. You are given the range that the number is in and then told "higher" or "lower" after each guess until you succeed in guessing it or run out of guesses. The Python program creates and holds the initial random number, and it is up to you (the user) to guess correctly. Use variables or constants to make it simple to extend to a different range and have more potential guesses.

Fundamental understanding of the Python programming.

For the Python guessing game.

- Create a flow chart.
 - Create a visual representation of the program's logic using a flowchart.
 - Clearly indicate decision points, loops, and the flow of the program.
- Create pseudocode.
 - Develop a pseudocode outlining the step-by-step logic of the program.
 - Include key variables, loops, and decision structures

➤ Create the python program that includes comments using # symbol.

Highlight the following features in your code.

- Indenting
- Repeat loop.
- Selection
- Colons
- Functions
- Text strings


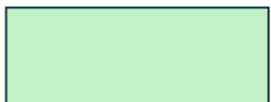

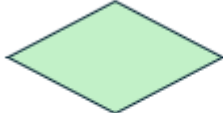

Assessment Criteria		
Grade	Description	Mark Range
Outstanding (O)	<ul style="list-style-type: none"> • A well-designed flowchart that clearly illustrates the program's logic. • Comprehensive and accurate pseudocode that effectively outlines the program's steps. • Python code is well-structured, readable, and follows best practices meeting all the requirements of scenario and working properly with no error and properly commented . 	45-50
High (H)	<ul style="list-style-type: none"> • A clear and logically structured flowchart. • Well-developed pseudocode, though with minor omissions or inaccuracies. • Python code is mostly well-organised with a few minor issues, however meeting all the requirements of scenario and working properly with no error and properly commented . 	40-44
Sound (S)	<ul style="list-style-type: none"> • A somewhat clear flowchart with some omissions or inaccuracies. • Adequate pseudocode but lacking in detail or clarity. • Python code is functional but may lack organisation or clarity in some sections. 	30-39
Basic (B)	<ul style="list-style-type: none"> • Flowchart is unclear or significantly lacks detail. • Pseudocode is incomplete or lacks clarity. • Python code has major issues in structure, readability, or logic. 	15-29
Limited (L)	Flowchart and pseudocode are severely lacking or incorrect.	0-14

Satisfactory completion of courses

A course has been satisfactorily completed, when the student has:

- Followed the course developed/endorsed by the NSW Educational Standards Authority (NESA)
- Applied himself/herself with diligence and sustained effort to the set tasks and experiences provided in the course.
- Achieved some or all of the course outcomes

Scaffold**Flowchart Scaffold**

Symbol	Name	Function
	Oval	Represents the start or end of a process
	Rectangle	Denotes a process or operation step
	Arrow	Indicates the flow between steps
	Diamond	Signifies a point requiring a yes/no
	Parallelogram	Used for input or output operations

1. Define the Purpose

- **Objective:** Identify the process or decision you want to illustrate.
- **Scope:** Determine the start and end points of the flowchart.

2. Identify Key Components

- **Start/End Points:** Represented by ovals or rounded rectangles.
- **Processes/Actions:** Represented by rectangles.
- **Decisions:** Represented by diamonds.
- **Inputs/Outputs:** Represented by parallelograms.
- **Arrows:** Indicate the flow of the process.

3. Outline the Steps

- **List Actions:** Enumerate each step or decision in the process.
- **Order of Operations:** Determine the sequence in which the steps occur.

4. Create the Flowchart

- **Start/End Symbols:** Place the start symbol at the top or beginning of your flowchart.
- **Process Steps:** Add rectangles for each step in the process.
- **Decision Points:** Insert diamonds where decisions are required.
- **Inputs/Outputs:** Use parallelograms to represent inputs or outputs.
- **Connectors:** Use arrows to show the direction of the flow between steps.

5. Review and Refine

- **Check for Completeness:** Ensure all steps and decisions are included.
- **Clarity:** Make sure the flowchart is easy to read and understand.
- **Consistency:** Ensure symbols and flow are consistent throughout the chart.

6. Finalize

- **Labels:** Add descriptive labels to each symbol.
- **Formatting:** Adjust layout for readability.
- **Feedback:** Get input from others to ensure accuracy and

Pseudocode Scaffold

PSEUDOCODE COMMON WORDS

1. **//**: This keyword used to represent a comment.
2. **BEGIN,END**: Begin is the first statement and end is the last statement.
3. **INPUT, GET, READ**: The keyword is used to inputting data.
4. **COMPUTE, CALCULATE**: used for calculation of the result of the given expression.
5. **ADD, SUBTRACT, INITIALIZE** used for addition, subtraction and initialization.
6. **OUTPUT, PRINT, DISPLAY**: It is used to display the output of the program.
7. **IF, ELSE, ENDIF**: used to make decision.
8. **WHILE, ENDWHILE**: used for iterative statements.
9. **FOR, ENDFOR**: Another iterative incremented/decremented tested automatically.

Steps involved in writing pseudocode

1. Define the Problem

Objective: Clearly state what the algorithm or process is intended to accomplish.

Inputs: List any inputs or initial conditions required.

Outputs: Specify what the algorithm should produce as output.

2. Outline the Steps

Initialization: Set up any initial variables or conditions.

Process Steps: Describe each step or action in the algorithm.

Decisions: Include conditions and branching (if applicable).

Loops: Describe repetitive actions if needed.

3 Review and Refine

Clarity: Ensure each step is clear and unambiguous.

Consistency: Use consistent terminology and formatting.

Completeness: Verify that all necessary steps are included.

4 Finalize

Format: Make sure the pseudocode is easy to read.

Test: Walk through the pseudocode to ensure it correctly represents the process.

Python code.

Your code should be indented and commented properly. Please refer to the PPT on canvas for more information

MARKING CRITERIA:

<p>Flowchart</p>	<ul style="list-style-type: none"> •A well-designed flowchart that clearly illustrates the program's logic covering all the necessary parts of your game. <p>(15 marks)</p>	<ul style="list-style-type: none"> • A clear and logically structured flowchart addressing all the necessary parts of your game. <p>(13 marks)</p>	<ul style="list-style-type: none"> •A somewhat clear flowchart with some omissions or inaccuracies. <p>(10 marks)</p>	<p>Flowchart is unclear or significantly lacks detail.</p> <p>(6 marks)</p>	<ul style="list-style-type: none"> •Flowchart and pseudocode are severely lacking or incorrect. <p>(3 marks)</p>
<p>Pseudocode</p>	<ul style="list-style-type: none"> • Comprehensive and accurate pseudocode that effectively outlines the program's steps. <p>(10 marks)</p>	<ul style="list-style-type: none"> •Well-developed pseudocode, though with minor omissions or inaccuracies. <p>(8 marks)</p>	<ul style="list-style-type: none"> •A somewhat clear flowchart with some omissions or inaccuracies. <p>(6 marks)</p>	<ul style="list-style-type: none"> •Pseudocode is incomplete or lacks clarity. <p>(4 marks)</p>	<ul style="list-style-type: none"> •Pseudocode is incomplete or lacks clarity. <p>(2 marks)</p>
<p>Python program</p>	<ul style="list-style-type: none"> • Python code is well-structured, readable, and follows best practices meeting all the requirements of scenario and working properly with no error and properly commented . <p>(25 Marks)</p>	<p>Python code is mostly well-organised with a few minor issues, however meeting all requirements of scenario and working properly with no error and properly commented .</p> <p>(23 Marks)</p>	<p>Python code is functional but may lack organisation or clarity in some sections.</p> <p>(15 Marks)</p>	<p>Python code has major issues in structure, readability, or logic.</p> <p>(10marks)</p>	<p>Python code is non-functional ,contains critical errors.</p> <p>(5 mark)</p>

		(22 marks)			
				TOTAL	/50