St Edward's College Malta

Year 10 Formal SBA 2026

LO13 Python Programming Task 2

Full Name

- This work is graded and contributes toward your SBA.
- The level of this task meets the criteria for SEC 09 Computing School-Based Assessment.
- The Assessment Rubric targets learing objectives at Level 1-2-3.
- The maximum mark awarded for this work is **30 marks**.

For teacher's use only:

Ms E. Camilleri

Criteria	**1**	**2**	**3**	**Total**
Obtained				
Allotted	**21**	**6**	**3**	**30**

Assessment rubric:

Assessment Rubric Task 2		
1. Python Programming Skills		
Variables for health, location, inventory, and game over condition.	**3**	
Output messages for instructions and descriptive game play.	**3**	
Arithmetic calculations for updating the player's health variable.	**3**	
Handling user choice when taking actions inside locations.	**3**	
Logic of the riddle guess and updating the game.	**3**	
Managing the list of inventory items (adding and checking items).	**3**	
Correct use of the main game loop to control game flow.	**3**	
2. Programming Practices		
Neat code that is easy to read.	**3**	
Able to debug or troubleshoot errors without help.	**3**	
3. Bonus Marks		
No tokens were used and/or creative enhancements included and/or the program is complete and fully functional.	**3**	
Total	**30**	

Marking Guidelines

0 – Fails to meet the criteria or not attempted.

1 – Was able to complete the task by following instructions.

2 – Was able to complete the task with limited support.

3 – Was able to complete the task independently.

Task 2: The Magic Door – Text Adventure Game

You will create a text-based adventure game in Python.

The player starts in a mysterious hall with three magic doors. Each door leads to a different location. By exploring the locations, the player can collect items, solve a riddle, and eventually win the game.

The game continues to run in a loop until the player either:

- wins the game, or
- loses all their health.

Game Flow Summary

- The player starts in the hall.
- The Green Door leads to a forest where a clue item can be collected.
- The Yellow Door leads to a cave where the player may fight a creature by guessing a riddle.
- If the riddle is guessed correctly, the player collects a key.
- The Purple Door leads to a treasure chamber.
- The player wins only if the key has been collected.

Program Requirements

Your program must:

- ask the player to enter their name
- use variables to store: player health, current location and game over condition
- use a list to store inventory items
- run inside a loop until the game ends
- use conditional statements to:
    - handle different locations
    - handle player choices
    - check inventory items
- include a riddle that affects the outcome of the game

Appendix 1: Teacher Help Tokens

You have **TWO** help tokens for coding hurdles.

Having tokens helps balance your independence while providing support so that you can be successful. Tokens are not used to get answers, but to get guidance on how to move on, for example, pointing you to relevant material or helping you get organized. Should you manage to complete the task to a high standard without any use of tokens, you get awarded bonus marks!

To use a token, raise your hand, and when the teacher approaches you, redeem a single token.

Help Token	**Help Token**
✋	✋

Appendix 2: Python Code Constructs

- Comment: `# This is a comment`

- Output:

Example 1

```
print(“This is a message”)
```

Example 2

```
print(message)
```

Example 3

```
print(“Message:”, message)
```

- Input: `user_name = input(“Enter name:”)`

- Variables and arithmetic:

Assignment (=)	`a = 0`
Addition (+)	`a = a + 1`
Subtraction (-)	`a = a - 1`

- Conditional operators:

Equal to	`a == b`
Not equal to	`a != b`

- Logical operators: `and`, `or`, `not`

- Conditional statement:

Example 1

```
x = 2
if x == 2:
    print("x is two!")
else:
    print("x is not two.")
```

Example 2

```
x = 2
if x == 1:
    print("x is one")
elif x == 2:
    print("x is two")
else:
    print("x is not valid!")
```

Example 3

```
name = "John"
age = 23
if name == "John" and age == 23:
    print("Ok, verified.")
else:
    print("Unverified.")
```

- Infinite loop with break:

```
while True:
    if condition == True:
       break
```

- Creating an empty list and adding items:

```
my_list = []
my_list.append("Jack")
```

- Check whether something is/is not in the list:

```
if "Jack" in my_list:
    print("Ok. Jack is on my list.")

if "Jack" not in my_list:
    print("Jack is not on my list.")
```

-- End of SBA --