# Assessment task

Year 12 ATAR Computer Science

## Task 2 - Unit 3

**Assessment type:** Practical test

**Conditions:** A 50-minute single lesson to complete the task

**Task weighting:** 5% of the school mark for this pair of units

**Total:** 40 marks

**Question 1 (5 marks)**

The UV index describes daily risk relative to the intensity of the sun's UV radiation. If it’s less than three, it’s safe. If it’s less than eight, protection is required. If it’s eight and above, you shouldn’t be out in the sun.

The program below requests the current UV rating for any city from an Application Programming Interface (API) and provides advice for users.

In the code provided, there is a syntax and a logic error. Correct those errors.

APIs are not always available due to server and network outages. Add an exception to this code to let the user know to try again later.

import requests

def main():

location = input("Enter a city to check your UV: ")

url = "https://wttr.in/{location}?format=%u\n"

response = requests.get(url)

uv = int(response.text)

if uv <= 2:

print("You can safely enjoy being outside!")

elif uv <= 7:

print("Seek shade! Slip on a shirt, slop on sunscreen and slap on a hat!")

else:

print("Avoid being outside! Make sure you seek shade! Shirt, sunscreen, and hat are a must!")

main()

The errors were that there was a missing colon in the if uv <= 2 statement, the other error was that the elif statement did not equal less than or equal to 7, instead it was more than or equal to.

**Question 2 (15 marks)**

A primary school teacher wants a program that provides students with 10 random multiplication questions. You must write this modular program.

When the program starts, it immediately shows the first question to the student. For example:

What is 9 x 5?

The program will let the student know if they got the question right or wrong. For example:

What is 9 x 5? 40

That is incorrect

What is 4 x 4? 16

That is correct

Once the 10 questions have been asked, it should show how many questions the student got wrong and what they were, so the student can show the teacher. For example:

There were 2 incorrect answers

What is 9 x 5?

What is 2 x 4?

Use a list to store the questions the user gets wrong.

You will need to import the random library:

import random

To generate random numbers use:

# returns a random number from 1 to 10

random.randint(1,10)

import random

incorrect\_questions = []

def\_create\_problem(num1, num2)

problem = print(f”What is {num1} x {num2}?”)

return problem

def\_answer(num1, num2):

answer = num1 \* num2

return answer

def\_questions():

score = 0

for i in range (num\_problems):

num1 = random.randint(1, 10)

num2 = random.randint(1, 10)

problem = create\_problem(num1, num2)

user\_answer = float(input(problem))

if user\_answer = solution:

print(“You’re right”)

score += 1

else:

print(“You’re wrong”)

incorrect.append(problem)

print(f“Your score was {score}.”)

if incorrect:

print(“These are the questions you got wrong:”)

for problem in incorrect:

print(problem)

def main():

questions()

main()

**Question 3 (17 marks)**

Create a Class called Student.

The Student class must have the following attributes: name, course name and an array for marks.

Create an instance of the Student class using the following details, name it student1:

"Mary", "Computer Science", [80, 90, 70, 85, 92, 88]

Create a method in the Student class to return the average mark for a student.

In main() use the student1 object to create the following output:

Mary’s average mark in Computer Science is: 84%

Create a method that returns a student's highest mark. Call your method to output the following:

Mary’s highest mark: 92

Add a method to the Student class to allow additional marks to be added to the marks array. Before adding the mark to the array, validate the mark value. Marks must be 0 to 100.

Create code in main() to have the end user enter 50 as a mark for Mary.

In main() output Mary’s details again.

Mary’s average mark in Computer Science is: 79%

class Student:

def \_\_init\_\_(self, name, course\_name):

self.name = name

self.course\_name = course\_name

self.marks = []

def calculate\_average(self):

average = sum(self.marks)/ len(self.marks)

return average

def highest\_mark(self):

highest = max(self.marks)

return highest

def add\_marks(self, mark):

if 0 <= mark <= 100:

self.marks.append(mark)

else:

print(“Invalid marks, must be between 0-100.”)

def main():

student1 = Student(“Mary”, “Computer Science”, [80, 90, 70, 85, 92, 88])

print(f”{student1.name}’s highest mark: {student1.highest\_mark()}”)

print(f”{student1.name}’s average mark in {student1.course\_name} is: {student1.calculate\_average}%”)

student1.add\_marks()

print(f”{student1.name}’s average mark in {student1.course\_name} is: {student1.calculate\_average}%”)

main()