**Analysing Day 17**

In Today’s day we made a Quiz game that used object oriented programming.

1. We analysed the data we are using to make a quiz game in the python file called data.py. We found it to be using the following structure a variable called question\_data that has a list of dictionaries with keys “text” and “answer”
2. Knowing that information we created a Class datatype to hold the data in a python file called question\_model which we made our first Class called Question which we then created a default constructor method using pythons \_\_init\_\_ class constructor. knowing from step one we created the method like this

**Def \_\_init\_\_ (self, qtext, qanswer) # which was the key variables from the dictionary**

We simply added attributes to the Question class like this calling self to signify this object

**Self.text = qtext**

**Self.answer = qanswer**

1. Creating a class structure that can hold the information from the database we call that Class in our main.py file calling it using this syntax **from question\_model import Question**
2. We must also call the data file in main.py using the following syntax **from** **data import question\_data**
3. Now in our main.py file we do the following: create a list called **question\_bank = []** which will hold an empty list we will use as a datatype to hold our Question Class.
4. Now we need to loop through the question\_data using the following syntax

**For question in question\_data:** #which will loop through our data

Create new variables which will hold values from the dictionary that is inside the list using syntax like this

**question\_ text = question[“text”]**

**question\_answer = question[“answer”]**

Then we will take those values and add initialise a new class object called with the following syntax **new\_question = Question(question\_text, question\_answer)**

Finally in our loop function we will append the question datatype into the list using the following syntax

**question\_bank.append(new\_quesiton)**

7. We will now create a new class file called quiz\_brain.py which will hold all the logic we can do in the question class

8. Creating a class using the same model as question we add a default constructor with the following attributes

Def \_\_init\_\_ (self, qlist) #which is that question list we made before

self.question\_number = 0

self.score = 0

self.question\_list = q\_list

9. Now creating a hierarchical class this function will implement most of the game logic

10. Now we will import into main.py using this syntax from quiz\_brain import QuizBrain and then we will create a object called quiz = QuizBrain(question\_bank)

11. Now we need to think about how we make game logic. How will the question keep iterating we must need to know if there is another question

So create a first method in QuizBrain called still\_has\_question

Writing it like this

def still\_has\_question(self):

return self.question\_number < len(self.question\_list)

12. Now in the main.py we can make a while loop that does

while quiz.still\_has\_questions():

13. Now we need to go to the next question and do a bunch of instructions which we will do by creating another method in QuizBrain called next\_question(self)

def next\_question(self):

current\_question = self.question\_list[self.question\_number]

self.question\_number += 1

user\_answer = input(f"Q.{self.question\_number}: {current\_question.text}. (True/False)?: ")

self.check\_answer(user\_answer, current\_question.answer)

Which will go to the current question and then grab user\_input in a variable and go to another QuizBrain method we need to create call check\_answer

14. We will implement it like this

def check\_answer(self, user\_answer, correct\_answer):

if user\_answer.lower() == correct\_answer.lower():

self.score += 1

print("Your got it right!")

else:

print("That's wrong.")

print(f"The correct answer was: {correct\_answer}.")

print(f"You're current score is {self.score}/{self.question\_number}")

print("\n")

Which will increase the attribute of score if gotten right of print out if you got it wrong

15. Finally in main.py we will print the quiz results used from the quiz brain object as the last instructions for the program

**Reflection on Day 17**