ICT/ Computer Science Department

KS4/KS5 LESSON PLAN

Topic / area of study: Python

Class: Year 10

Ability: Mixed ability.

Date & period: 24/11/2017 Lesson 4

Learning aim:

Pupils to advance their knowledge on strings by learning some string operations as well as concatenation and converting to string. They will also learn how to take input from the Python shell. After this, basic IF statements will be taught and then finish with an exercise that tests all the knowledge learnt in the lesson.

Learning objectives/Success criteria:

All pupils will:
Complete the first three questions in exercise 2.
Find two syntax errors in code.
Program and understand advanced strings.
Most pupils will:
Complete all questions in exercise 2.
Find three syntax errors in code.
Program and understand input function.
Some pupils will:
Complete an extension task whether it is from the booklet or a different one mentioned in MAT section.
Find four syntax errors in code.

Lesson starter / activate the learner

5 minutes

The start of the lesson will be used to recap the previous lesson, specifically the exercises. The teacher will go through each question in exercise 1 quickly to refresh pupils' memory and clear up any problems they may have had. Solutions for the questions will be on the board. This will involve praising the class for their effort and progress in last lesson. It will be interactive with the teacher asking questions and allowing pupils to give answers. It also gives pupils an opportunity to confirm their answers or correct them by writing down the code.

Main Activity

10 minutes = explaining advanced strings and input

Get straight into the content by asking a few questions about strings such as "Does anyone know any string operations in Python?" get a feel for the classes knowledge in this area, it should be minimal. Make sure they are on [page 5] of the booklet as it has information on string operations and concatenation in print statements. Expand on the information such as "Do you think the first print statement on page 5 is good practice", ask for reasoning, state that there should be a space in between 'firstName' and 'lastName'. This will keep the pupils thinking and focused. Explain upper(), lower() and len() but ask pupils first and then reiterate the point for the class. Give the class 30 seconds to fill in the three lines on [page 6]. Ask questions about user input to identify any prior knowledge, if a pupil is confident ask them to explain to the class and then reiterate the point. Add any information not covered yet such as you can also print to the Python shell as well as waiting for user input. Refer the class to the question on [page 5] that says "Why assign it to a variable?", after some discussion reach the point that it's because we want to store the user input in order to use it.

5 minute = debugging activity

Briefly explain what debugging means and how it is very important and then put the Python code on the board that has syntax errors in it. The class have not had enough experience to provide code with logical errors, which is why there are only syntax errors, four to be exact. The code focuses on skills just learnt in the lesson. The class should work in pairs and will have three minutes to complete it with one minute remaining for the teacher to go over the four errors. The file has the solutions in comments below so make sure to hide them or remove before putting it on the board. This activity will improve the pupils team working skills as well as problem solving skills.

10 minutes = explaining IF statements

Start by asking the class if they know what an IF statement is and start to engage with a few pupils. Then proceed to explaining about what they are, how they are commonly used and how they allow programmers to make decisions. Refer to the booklet on [page 5] as it shows the syntax and shows a diagram representing a decision. Expand on this and write the structure of an IF statement in pseudocode on the board, showing exactly where the condition and action would go. To ensure the class are following and understand the concept of IF statements there will be an exercise involving the whole class. The exercise involves stating a condition with an action as well as another condition and action. The teacher will choose conditions that will get the majority of the class involved if not all such as 'hands up if you're wearing a jumper'. This will be followed by one or two live coding examples to demonstrate the syntax of IF statements to the class, depending on time. During the examples the teacher will link back to the class examples such as "if you are wearing a jumper" condition in order to demonstrate where they would fit in the structure of an IF statement.

Main Activity (continued)

25 minutes = exercise that involves all programming skills learnt in the lesson

There are four questions in exercise 2, which is on [page 6]. Two of the questions require some writing in the booklets, they are included since programming does not involve any writing skills. Therefore, they have been included in exercises when possible. Refer the pupils to [page 8] as question 4 requires a new operator but does not require teaching. Observe if the pupils can work it out for themselves. The teacher should identify the MAT pupils and try not to support them at all, divert the pupils to google their problems since it is a good skill to acquire. Pupils will be expected and told to save each task a separate file inside a folder named 'Exercise 2' in order to check what extent pupils have successfully achieved the learning objectives. However, it will not be marked properly. For information on extension tasks go the MAT section of the plan. At the end, make sure the class has saved their work.

Plenary

5 minutes

This section is to review the learning taken place in the lesson. Ask specific questions such as "How did you find using the input() function in Python?". The pupils can answer with either thumbs up which indicates well understood, thumbs down which indicates didn't understand and thumbs sideways to indicate somewhere in between. This will give the teacher feedback in terms of the difficulty of the lesson and improve the pupils learning, as they are required to recall the knowledge from their brains. It may even help pupils who were finding something difficult. Test as many pupils as possible with elements learnt in the lesson such as "What do I need to do in order to use input from the shell later on?". Lastly, let the pupils know that they are having a written exam and suggest looking over their work booklet and Python files. Inform the pupils they will have half an hour.

Content of preceding lesson (if applicable)

Recap knowledge on using the Python idle. Covered data types and variables with an exercise to put their knowledge into practice.

Content of lesson to follow

Covering more advanced IF statements that include an else branch with a brief exercise. This will be followed by a test to see how well the pupils are achieving.

Essential/Key Skills covered

Digital competence:
How to manipulate strings in Python, convert data types and take input from the Python shell.
Program and understand simple IF statements.
First experience of fixing code the pupils have not seen before improving their debugging skills.
Curriculum Cymreig: Greetings and other commands in welsh, page numbers in the workbook have welsh next to the number.
Numeracy: Exercise 2 Q2 and the two extension tasks.
Literacy: Exercise 2 Q3 requires an answer in full sentences. There is also reading required on page 5.

Additional information

Describe how the lesson addresses the progress and learning of the following groups of pupils:

EAL (English as an additional language)

The lesson provides a lot of visuals as well as verbal. Teacher will provide more support to them by repeating the instructions once the class are doing something practical, as well as reading out the exercise questions. Lastly, the teacher will make a conscious effort to speak clearly and slower, using basic key phrases and words to allow EAL pupils to have a better chance of understanding. Suggest the pupil sitting at the front in order to have a better chance of understanding.

ALN (Additional Learning Needs)

Provide as much support as possible. The lesson is planned with lots of different ways of learning in the hope that it will be useful for all groups of pupils. There are many questions to interact and engage with pupils as well as an interactive exercise with variables. There are live examples to demonstrate visually as well as verbally. Also, the questions are designed to suit all pupils and slowly progress.

MAT (More, Able and Talented)

If pupils finish exercise 2 on [page 6] fast then they will be challenged to create a program that prompts a user to input a year and then check if it is a leap year. This is shown underneath the exercise on [page 6]. It will require nested IF statements that they have not done before. The teacher can tell them what they need but not how to do it. The concept of checking a leap year is explained on [page 6] If pupils do not complete it, they will be encouraged to try it at home. If the MAT pupils find this too challenging then they can create a program that checks if a number is even or odd. The program is slightly more difficult than the exercise as they need to use modulus but it is easier than the lea year program. There are solutions for both of these programs if a pupil requires. Failing both of those an online resource [1]has been provided which has tutorials on IF statements for extra learning. The other resources referenced in lesson plan 1 had IF statement sections but were too difficult for year 10.

Additional Information

[2] This idea involves pupils sabotaging their partners work and then the other partner tries to fix it. This is a really good idea but is quite a long process. Therefore, I adapted it to me creating the code and two pupils work together to fix it instead of one. Also, the sabotaging part has been removed. If there were extra time, more of this idea would be implemented into the plan. The only resources required are the work booklets for the children to follow and answer in as well as computers for the pupils to program on. Lastly, a projector and board will be needed in order to demonstrate examples.

In terms of health and safety, ensure the bags are under their tables before sitting at the computers in case pupils were to trip over a bag. Do not allow pupils to swing on their chairs and ensure the pupils do not put any liquids near the computers such as bottles of water. If a pupil is too close to the screen, ask them to sit back a little.

To enforce curriculum cymreig greeting at the beginning of the lesson should be in welsh. Also try to reiterate other commands such as "dim siarad" when asking for quiet in the classroom.

References
[1] “learnpython”, Variables and types, available at:
http://learnpython.org/en/Conditions

[2] “Sabotage”, Blog involving a debugging activity available at:
https://teachcomputing.wordpress.com/2013/11/23/sabotage-teach-debugging-by-stealth/

Solutions for Exercise 2

Q1)
```
day = 04
month = 12
year = 2017

dob = str(day) + "/" + str(month) + "/" + str(year)
print(dob)
```

Q2)
```
answer = input("What is 4 x 4?")
print("The answer is " + str(answer))
```

Q3)
```
mark = input("What mark did you receive in coursework?")
if mark > 60:
    print("Well done!")
```

Q4)
```
friend = raw_input("What is your friends name?")
friend = friend.lower()
if friend == "william":
    print("william loves Python")
if friend != "william":
    print(friend,"loves football!")
```

Leap Year Extension Solution

```
year = int(input("Enter a year: "))
if (year % 4) == 0:
   if (year % 100) != 0 or (year % 400) == 0:
       print(year,"is a leap year")
```

Odd or Even Extension Solution

```
num = input("Enter a number: ")
if num % 2 == 0:
    print(num,"is an even number")
else:
    print(num,"is an odd number")
```

Syntax Error Code and Solutions in Comments

```
firstName == input("What is your first name?")
lastName = input("What is your last name?)
print(firstName + " " + lastName)
age = 17
print(firstName + " is" + age + " years old!")
print("The length of your last name is",lastName.len())

# line1 - == should be =
# line2 - missing " after the question mark
# line5 - cant do + age needs to be , age because its not a string
# line6 - need to use length function like len(lastName)
```