Computing

Lesson 3: For Loops

Programming Part 3: Iteration

Rebecca Franks

[1] *Materials from the Teach Computing Curriculum created by the National Centre for Computing Education*

Times table generator

Task 1: Predict

Take a look at the code below. Read it carefully and try to make a prediction about the output of this program when it is executed.

```
1  times_table = 5
2  answer = 0
3  print(f"Here is the {times_table} times table")
4  for x in range(1,11):
5      answer = x * times_table
6      print(f"{x} times {times_table} is {answer}")
```

Remember to write your prediction down!

Task 2: Run

Open and run the file in Repl.it **oaknat.uk/comp-ks4-timestable**

- Was your prediction correct?
- Did anything unexpected happen?
- Write down your thoughts.

Task 3: Investigate

Investigate the program using the steps below:

Step 1

Execute the code.

- What is the **2nd line** that is output for display?
- What is the last line that is output for display?

Step 2

Line 6 is used to output the times table message on the screen.

- In the first iteration, what is the value of `x`?

Step 3

Investigate the variable `times_table`.

- Does the value of `times_table` change throughout the running of the code?

Task 3: Investigate

Investigate the program using the steps below:

Step 4

Investigate the variable `answer`

- What arithmetic expression is assigned to `answer`?

Step 5

Execute the code.

- What happens to the value of `x` after each iteration?

Step 6

In the final output, what is the value of `x`?

Task 3: Investigate

Investigate the program using the steps below:

Step 7

Line 4 has the code `range(1,11)`

- Change the values to `range(2,22)`
- What is the 2nd line that is output for display?
- What is the final line that is output for display?

Step 8

Change the range to `range(10,30)`

- What is the 2nd line that is output for display?
- What is the final line that is output for display?

Step 9

How do you think the value `x` relates to the values passed in `range`?

Task 4: Modify

Modification 1	Hint

Change the values passed in range so that the times table generator will output the 5 times table from 1 to 12.	The 2nd line should output as:
	`1 times 5 is 5`
	
	The final line should output as:
	`12 times 5 is 60`

Modification 2	**Hint**
Introduce an `input()` to the program to allow the user to enter the times table that they wish to create.	Remember to include a message to the user before they enter the input.

Task 4: Modify

Modification 3	Hint

Test your code.
 	
If the user enters a 6 it should display the 6 times table.
	Expected output:
	
	`Welcome to the times table generator`
	`Enter a times table that you would like to display:`
	`6`
	`Here is the 6 times table`
	`1 times 6 is 6`
	`2 times 6 is 12`
	`3 times 6 is 18`
	`4 times 6 is 24`
	
	`etc...`

Task 4: Modify

Modification 3	Hint

Continued...	**Unexpected output:**
	
	If your output is like the example below then think about what needs to be added to your code to stop the program using the value as **string** (text).
	
	`Welcome to the times table generator`
	`Enter a times table that you would like to display:`
	`6`
	`Here is the 6 times table`
	`1 times 6 is 6`
	`2 times 6 is 66`
	`3 times 6 is 666`
	`4 times 6 is 6666`
	
	`etc...`

Task 4: Modify

Modification 4	Hint

Add some additional functionality to your code by allowing the user to enter the value that they wish the times table to go up to.
 	
• Create a variable called `max_value`
• Assign a value to `max_value` through user input.
• Add a message prior to the `max_value` input to let your user know what they need to enter.
• Replace 13 in the `range(1,13)` with `max_value`
• Test your code.
	Expected output if 6 and 10 are input:
	
	`Welcome to the times table generator`
	`Enter a times table that you would like to display:`
	`6`
	`Enter the maximum value for your times table:`
	`10`
	`Here is the 6 times table`
	`1 times 6 is 6`
	`2 times 6 is 12`
	`3 times 6 is 18`
	`4 times 6 is 24`
	`5 times 6 is 30`
	`6 times 6 is 36`
	`7 times 6 is 42`
	`8 times 6 is 48`
	`9 times 6 is 54`

Task 4: Modify

Modification 5	Hint

You will notice that the maximum value entered is not reached when this code is run because of how the range function works.
 	
● Directly below the code where the `max_value` is input, enter some code that will increase that value by 1.
● Test your code.
	Expected output if 6 and 10 are input:
	
	`Welcome to the times table generator`
	`Enter a times table that you would like to display:`
	`6`
	`Enter the maximum value for your times table:`
	`10`
	`Here is the 6 times table`
	`1 times 6 is 6`
	`2 times 6 is 12`
	`3 times 6 is 18`
	`4 times 6 is 24`
	`5 times 6 is 30`
	`6 times 6 is 36`
	`7 times 6 is 42`
	`8 times 6 is 48`
	`9 times 6 is 54`
	`10 times 6 is 60`

Task 4: Modify

Modification 6	Hint

Use `try:` and `except:` to ensure that the input for `times_table` and `max_value` are input is an integer.	**Sample code from Lesson 4:**
	
	`print("Enter a number")`
	`try:`
	`    number = int(input())`
	`except ValueError:`
	`    print("You must enter a number")`
	`    number = int(input())`

Times table quiz

Times table quiz

Create a quiz that tests a players knowledge of their times tables. The program should:

- Ask the user which times table they would like to test.
- Ask the maximum value they would like to go to.
- Go through each multiplication in turn and ask for the correct answer.
- If the player is correct then correct should be displayed.
- Else incorrect should be displayed.

Times table quiz

Step 1

Use your modified code from the previous task as your starting point. You might want to save a new copy and keep the other version as a backup.

Step 2

Place a tick ✓ next to the instruction when you have completed it.

- [ ] Modify the welcome message to greet the users to the times table quiz.
- [ ] Modify the print statements for the two inputs in reference to the quiz.
- [ ] Test your code.

Times table quiz

Example

Note: Use this example to check your program. Given the input you see in this sample interaction, this is the output your program should produce.

When the program is executed, it should display a similar prompt.	`Welcome to the times table quiz` `Enter a times table that you would like to be tested on:`
The user types their reply.	`7`
A second prompt appears.	`Enter the maximum value for your times table:`
The user types their reply.	`12`

Continues on next slide...

Times table quiz

Continued...

The original output for your previous program should display.

```
Here is the 7 times table
1 times 7 is 7
2 times 7 is 14
3 times 7 is 21
4 times 7 is 28
5 times 7 is 35
6 times 7 is 42
7 times 7 is 49
8 times 7 is 56
9 times 7 is 63
10 times 7 is 70
11 times 7 is 77
12 times 7 is 84
```

Times table quiz

Step 3

Just before the for loop is this line of code:

```
print(f"Here is the {times_table} times table")
```

- [ ] Rephrase the code to say that they will be tested on their chosen times table.

The user needs to make their guess for the correct answer.

- [ ] At the end of the for loop. Create a variable called user_answer and assign it an integer input.
- [ ] Add a user prompt to ask them for their answer.
- [ ] Test your code.

Example input/output for testing

Example

Note: Use this example to check your program. Given the input you see in this sample interaction, this is the output your program should produce.

When the program is executed, it should display a similar prompt.	`Welcome to the times table quiz` `Enter a times table that you would like to be tested on:`
The user types their reply.	7
A second prompt appears.	`Enter the maximum value for your times table:`
The user types their reply.	12

Continues on next slide...

Example input/output for testing

Continued…	`Here is your quiz on the 7 times table`
	`1 times 7 is 7`
	`Answer:`

The program displays a similar prompt.	
The user types their reply.	`7`
The program displays this prompt.	`2 times 7 is 14` `Answer:`
The user type their reply.	`14`

Note: this should continue until the end of the sequence is reached.

Times table quiz

Step 4

Place a tick ✓ next to the instruction when you have completed it.

- [ ] Under the variable that holds `user_answer`, create an `if-else` statement that will check if the user answer is the same as answer.
- [ ] If the user is correct it should display “`correct`” if they are incorrect it should display “`incorrect`”.
- [ ] Test your code.

Example input/output for testing

Example

Note: Use this example to check your program. Given the input you see in this sample interaction, this is the output your program should produce.

The user replies to two prompts.	7 12

A prompt is displayed.	`Here is your quiz on the 7 times table`
	`1 times 7 is 7`
	`Answer:`

The user types their reply.	7

A prompt is displayed.	`Correct`
	`2 times 7 is 14`
	`Answer:`

The user types their reply.	17

A prompt is displayed.	`Incorrect`
	`3 times 7 is 21`
	`Answer:`

Times table quiz

Place a tick ✓ next to the instruction when you have completed it.

- [ ] Adjust this line of code so that the answer is not revealed to the user:

```
print(f"{x} times {times_table} is {answer}")
```

- [ ] Test your code.

Note: the first question prompt should now look like this:

```
Here is your quiz on the 7 times table

1 times 7 is ...

Answer:
```