PROGRAMMING FUNDAMENTALS REVISION LECTURE

SOFT6018 – WEEK 10

LOOPS

A repetition structure causes a statement or set of statements to execute repeatedly.

- A *condition-controlled loop* uses a true/false condition to control the number of times that it repeats.

- A *count-controlled loop* repeats a specific number of times.

TURTLE MODULE

TURTLE GRAPHICS IS AN INTERESTING AND EASY WAY TO LEARN BASIC PROGRAMMING CONCEPTS. THE PYTHON TURTLE GRAPHICS SYSTEM SIMULATES A “TURTLE” THAT OBEYS COMMANDS TO DRAW SIMPLE GRAPHICS.

THE FIRST STEP IN USING PYTHON’S TURTLE GRAPHICS SYSTEM IS TO WRITE THE FOLLOWING STATEMENT:

```
import turtle
turtle.forward(200)
turtle.right(90)
turtle.forward(200)
```

PROBLEM DEFINITION

WE WILL START WITH A PROBLEM DEFINITION: WRITE A PROGRAM WHICH GIVES THE USER THE OPTION OF DRAWING A CIRCLE OR SQUARE OR RECTANGLE.

LET'S START WITH THE MENU THEN ADD THE DRAWINGS THEMSELVES, THEN ADAPT THAT CODE TO ASK THE USER FOR THE SIZE OF THE SHAPES.

MENU (IFS)

Ask the user if they want to draw a

1. Circle
2. Square
3. Rectangle

For now we will just print the word corresponding to their choice.

MENU (IFS)

Ask the user if they want to draw a

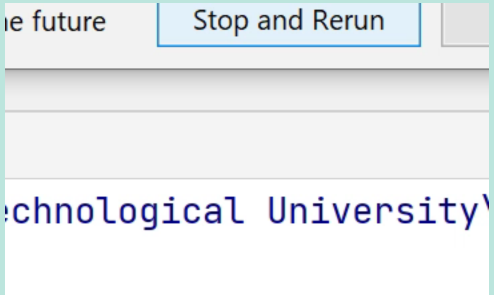
1. Circle (add code to draw a circle)

```
turtle.circle(100)
```

2. Square

3. Rectangle

For now we will just print the word corresponding to their choice.



```
chnological University
```

ADD SQUARE (LOOP)

This draws a line, turns right by 90°, draws another line, turns right by 90°, draws another line turns right by 90°, and draws another line, turns right by 90° to get back where it started.

So, in effect it draws a line and turns right by 90° four times – let's write a loop.

ADD RECTANGLE

This is very similar to a square, but the width and height differ from each other.

So, this time it draws a line of width, turns right by 90°, draws another line of height, turns right by 90°, draws another line of width, turns right by 90°, and draws another line of height, turns right by 90° to get back where it started.

ADD INPUTS

We will now ask the user for

- Circle – size of the radius
- Square – size of a side
- Rectangle – size of the width and size of the height

MANY CIRCLES

Let's say we want to draw several circles starting from the same point.

We will ask the user for the starting radius of the circle. We can draw circles reducing the radius by 5 each time until the radius is 0.

So we draw the circle the reduce radius by 5 then draw the circle, then reduce radius by 5 until the circles disappear.

MANY SQUARES (NESTED LOOP)

Let's draw several squares.

We will draw a square, reduce its size, draw it again, reduce its size and keep going until the size of the square is 0.

EXTRA EXERCISES

RANDOM AND TUPLES

I would like to change the colour. I could ask the user to choose a colour by name or from a menu but instead Python will randomly select an option from a collection of colours.

```
import turtle
import random


COLOURS = ("red", "green", "blue", "cyan", "purple", "yellow", "pink", "orange")

choice = int(input("Do you want \n1: circle\n2: square\n3: rectangle\n>>> "))

number_colours = len(COLOURS)
random_colour = random.randint(0, number_colours - 1)
turtle.pencolor(COLOURS[random_colour])
turtle.width(3)
```

MORE COLOUR (USE OF INDEX AND %)

Let's add some more colour by going through each colour in the tuple, than starting again if we run out of colours.

We need a counter which starts at 0, goes through each index and returns to 0 when it reaches the number of colours.

```
colour_index = colour_index + 1
```

moves to the next colour index

```
colour_index = colour_index % number_colours
```

will return to 0 when the number of colours is reached but any value less than the number of colours will be unchanged.

ASK THE USER TO CHOOSE A COLOUR (IF/IN WITH LOWER())

Let's ask the user to pick a colour. They will type a colour as a string.

If the colour is in the tuple it will be applied. Otherwise a random colour will be set.

```
side = int(input("Side >>> "))
colour = input("What colour would you like this to be? ")
if colour in COLOURS:
    turtle.color(colour)
else:
    random_colour = random.randint(0, number_colours - 1)
    turtle.pencolor(COLOURS[random_colour])
    print(f"A random colour of {COLOURS[random_colour]} was selected.")
```

But what if they type Pink instead of pink? Let's convert whatever the user types into lowercase using a string method

```
colour = input("What colour would you like this to be? “).lower()
```

LOGIN – FILES

We wish to limit the user of our program to only authenticated users who are at least 5 years of age.

Create a file with usernames, passwords and ages.

Ask the user for their username and password.

Determine if that username/password combination is in the file.

If it is in the file, determine if the user is old enough to use our program.

Passwords are case sensitive.

Usernames are not case sensitive.

This requires converting the age to an integer.

LOGIN – FILES

```
# Create menu
import turtle
import random
# If someone asks then try the following but know it only works in terminal so must run in terminal not
# import getpass
MIN_AGE = 5
users_username = input("Username: ").lower()
users_password = input("Password: ")
# users_password = getpass.getpass()
logged_in = False

users_file = open("users.txt")


# continued on the next slide
```

LOGIN – FILES

```
for line in users_file:
    line = line.strip()
    line = line.split(',')
    stored_username = line[0]
    stored_password = line[1]
    stored_age = int(line[2])
    if users_username == stored_username and users_password == stored_password

                        and stored_age >= MIN_AGE:

        logged_in = True

users_file.close()

if logged_in:

    # previous code goes here

else:

    print("You cannot use our program 😔 ")
```