ITI 1120
Lab # 1

An Introduction to the Lab Environment

About TA (Teaching Assistant) …

- Name:
- E-mail:

Objectives

- Getting familiar with the lab environment
    - logging in and logging Out
    - email
    - Brightspace
    - Coursera
- Creating and Submitting a Practice Assignment 0 on Brightspace
- Enrolling into two Python courses on Coursera
- Writing your first Python program
- Two Python programming exercises (with Turtle graphics)
- Blockly exercises

Logging in and Logging out

LOG IN:

Follow the instructions on the screen to log in. In particular:

•Use your uoACCESS accounts to authenticate, which is the same account used for Gmail, Google Docs and Eduroam. For example, if your student's email is jdoe123@uOttawa.ca, you will specify the username 'jdoe123' and use your uoACCESS password.

LOG OUT:

•When you are all done and ready to leave, the last thing you **ALWAYS** do is to log out. Another student could have access to your files if you do not log out. Do not turn the power off after logout, leave the power on.

University Email

Your university email is powered by Google.

To sign up for a Google Apps account, log into uoZone from the student portal
 http://uOttawa.ca/students
and click on Email management in Key applications on the right hand side of the homepage.

Accessing Brightspace

- ## What is at Brightspace?
  - All the course material is there (think of it as the course web page)
  - Downloading and submitting assignments
  - Announcements
  - Consulting your grades.
  - Discussion groups/forums
  - Students are required to check Brighhtspace course webpage and their OttawaU email frequently for announcements and all material.

- ## To access brightspace:
  - Go to https://uottawa.brightspace.com/d2l/home
  - Enter your user name and password and sign In.
  - Click on ITI1120 to access the course

Starting Lab 1

- Open a browser and log into Brightspace

- On the left hand side under Labs tab, find lab1 material contained in lab1-students.zip file

- Download that file to the Desktop and unzip it.

Starting Lab 1

- Open the file provided with this lab called OpeningIDLE.pdf

- Follow the 5 steps described in OpeningIDLE.pdf

In Python shell type:
- type 1+2*3, press enter and observe the output
- Then type (1+2)*3, press enter and observe the output
- Then type 2**10, press enter and observe the output
- Then type 11/2, press enter and observe the output
- Then type in the expression that sums first 5 positive integers
- Then type the expression that finds average of 25, 12, 40 and 1

Practice Submitting Assignment 0

- **First to learn how to create your submission follow the steps below:**

  – Open the file provided with this lab called UsingIDLE-first-program.pdf

  – Follow the 10 steps to create Python program A0_xxxxxxx. py and place it in the folder you crated called A0_xxxxxxx.
     (Note that what you see on your computers may differ slightly from that what you see in UsingIDLE-first-program.pdf)

  – Right-click on the A0_xxxxxxx folder and then select "Add to A0_xxxxxxx.zip".
    - This will create a zip file with the contents of the directory.

- **Then to submit (A0_xxxxxxx.zip) follow the steps below:**
  – Go to Brightspace uottawa.brightspace.com
  – On the left-hand side click on Assignments tab
  – Click on Assignment 0
  – Under "Submissions" you should see dashed rectangle labeled with "Drop files here, or click below!"
  – Drag and drop A0_xxxxxxx.zip there. You should now see A0_xxxxxxx.zip in that rectangle.
  – Scroll down and click on "Submit".
  – You will see a pop-up saying "+Submitted successfully". You will also receive a notification email.
  – You can submit more than once. Try it! The old submission will not be deleted.

Identification

- The following information must be included at the beginning of each program in your future assignments. For example, in Assignment 1, your program a1_xxxxxxx. py must start with:

```
# Course:   IT1 1120
# Assignment number
# Family name, Given name
# Student number
```

Finishing up with lab technicalities

• When you log out from a lab computers, the files and folders you created will disappear. Thus if you need them copy them to a USB memory stick before logging out.

• When you are ready to leave, double-click on the logout icon. Do not turn off the power.

• But, before logging out … complete the following tasks

Enroll into two Coursera courses

1. If you do not have Coursera account, first create one here and remember your login credentials:
   https://www.coursera.org/learn/learn-to-program?authMode=signup

2. Enroll into the following two free Python courses:
   (If there is some option to pay something – do not pay anything. Choose the free option.)

https://www.coursera.org/learn/learn-to-program

and

https://www.coursera.org/learn/program-code

Make sure you enroll now, as enrollment will not be possible after the posted enrollment deadline.

Turtle Program Example

- Start IDLE

- The material you downloaded for Lab 1 has a Python program in a file called `turtle_smiles.py`

- Open that file with IDLE

- Press Run Module and observe what happened.

- Now study the code and the following two slides before attempting to do Exercise 1 (explained in 3rd slides from here)

- Notice the lines in `turtle_smiles.py` that start with #. All lines starting with # will be ignored by python interpreter when your press run. Therefore I used # to explain parts of the program to you.

Useful functions in Turtle Class

Table CS.1 Some methods of the Turtle class.
After importing the module `turtle`, you can obtain the full list of Turtle methods in your interactive shell using `help(turtle.Turtle)`

| Usage	Explanation
`t.forward(distance)`	Move turtle in the direction the turtle is headed by distance pixels
`t.left(angle)`	Rotate turtle counterclockwise by `angle` degrees
`t.right(angle)`	Rotate turtle clockwise by `angle` degrees
`t.undo()`	Undo the previous move
`t.goto(x, y)`	Move turtle to coordinates defined by x and y; if pen is down, draw line
`t.setx(x)`	Set the turtle's first coordinate to x
`t.sety(y)`	Set the turtle's second coordinate to y
`t.setheading(angle)`	Set orientation of turtle to `angle`, given in degrees; Angle 0 means east, 90 is north, and so on
`t.circle(radius)`	Draw a circle with given `radius`; the center of the circle is `radius` pixels to the left of the turtle
`t.circle(radius, angle)`	Draw only the part the circle (see above) corresponding to `angle`
`t.dot(diameter, color)`	Draw a dot with given `diameter` and `color`
`t.penup()`	Pull pen up; no drawing when moving
`t.pendown()`	Put pen down; drawing when moving
`t.pensize(width)`	Set the pen line thickness to `width`
`t.pencolor(color)`	Set the pen color to color described by string `color`

Useful functions in Turtle Class

Usage	Explanation
s.bgcolor(color)	Changes the background color of screen s to color described by string color
s.clearscreen()	Clears screen s
s.turtles()	Returns the list of all turtles in the screen s
s.bye()	Closes the screen s window

Table CS.2 **Methods of the** Screen **class.** Shown are only some of the Screen class methods. After importing module turtle, you can obtain the full list of Screen methods in your interactive shell using help(turtle.Screen)

This table is from "Introduction to Computing Using Python" by Lj. Perkovic

15

Exercise 1

- From IDLE open file turtle-ex1.py that is provided with Lab 1.

- Add your own code inside of that file so that your final program draws a dart board similar to the one depicted below. (no need to use loops. copy/paste is your friend in this lab). Do not forget that whenever you make changes to a file, you have to save it first

  before you can run it.
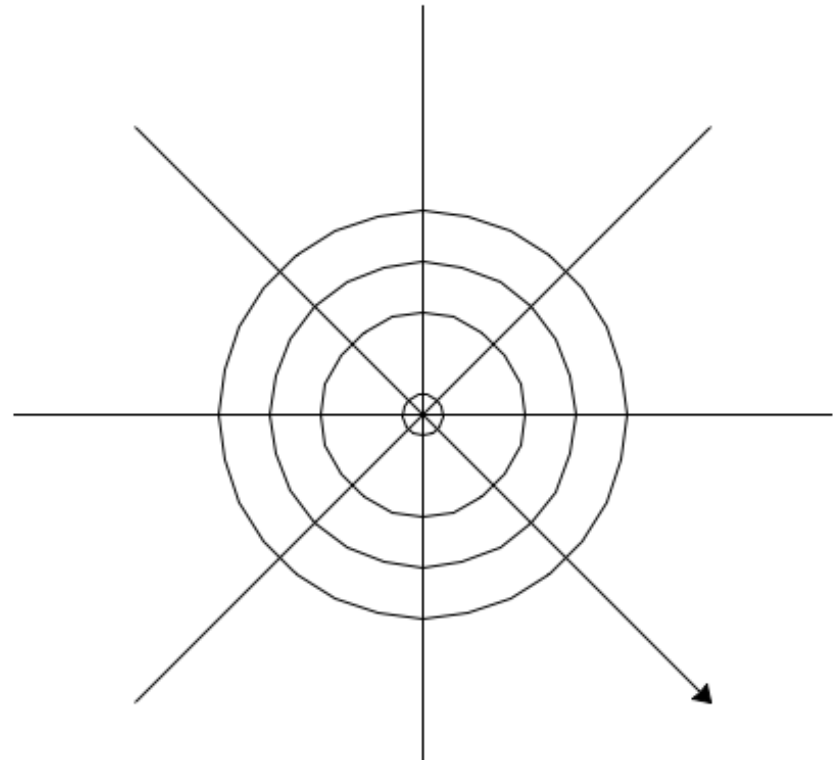
Exercise 2

- From IDLE open file turtle-ex2.py that is provided with Lab 1.
- Add your own code inside of that file so that your final program draws an image similar to the one depicted below. (no need to use loops. copy/paste is your friend in this lab). Place your code where indicated in file turtle-ex2.py
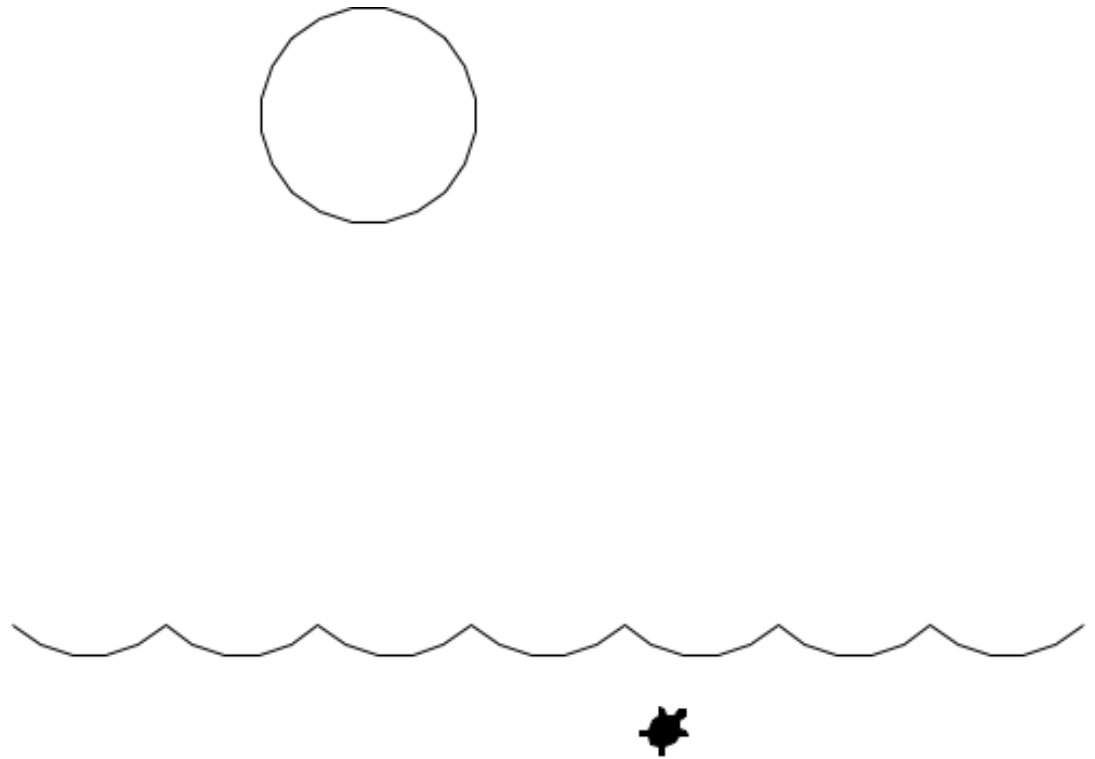
Having fun:
write a program in Blockly

- What is Blockly: visual programming language developed by Google (to help students learn programming)

- Instructions are blocks that fit together (like lego bricks) to make up a computer program.

Final Exercises

•Use Blockly to solve level 9 and 10 mazes here:

https://blockly-games.appspot.com/maze?lang=en&level=9&skin=0

https://blockly-games.appspot.com/maze?lang=en&level=10&skin=0

-To get an idea on how to solve them (i.e. how to program in Blockly, you many need to solve a few earlier levels)

-Try to find solutions with the smallest possible number of blocks

-Challenge: Can you solve level 9 with 4 blocks. What about level 10 with 5!!