

ITI 1120

Lab # 1

Introduction

# About TA (Teaching Assistant) ...

- Name:
- E-mail:

# Objectives

- Installing Python and starting IDLE
- Creating and Submitting a Practice Assignment 0 on Brightspace
- Enrolling into two Python courses on Coursera ([this only reason for this is so you get access to their Coursera quizzes for self-evaluation purposes](#))
- Writing your first Python program
- Two Python programming exercises (with Turtle graphics)
- Blockly exercises

# Accessing Brightspace

- What is at Brightspace?
  - All the course material is there (think of it as the course web page)
  - Downloading and submitting assignments
  - Downloading and submitting tests
  - Announcements
  - Consulting your grades.
  - Discussion groups/forums
  - Students are required to check Brightspace course webpage and their OttawaU email frequently for announcements and all material.
- To access brightspace:
  - Go to <https://uottawa.brightspace.com/d21/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 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 created 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](https://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_XXXXXX.py` must start with:

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

# Pause

At this point TA(s) will help anyone who has issues with: installing python, opening IDLE, running their first python program, or submitting Assignment 0

- Everyone else please continue with lab work as explained on the remaining slides.

# 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>

Again, the only purpose of enrolling in these 2 coursera courses is so that you get access to their quizzes for the self evaluation purposes.

# 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 3<sup>rd</sup> slides from here)
- Notice the lines in `turtle_smiles.py` that start with **#**. All lines starting with **#** will be ignored by python interpreter when you 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
<code>t.forward(distance)</code>	Move turtle in the direction the turtle is headed by <code>distance</code> pixels
<code>t.left(angle)</code>	Rotate turtle counterclockwise by <code>angle</code> degrees
<code>t.right(angle)</code>	Rotate turtle clockwise by <code>angle</code> degrees
<code>t.undo()</code>	Undo the previous move
<code>t.goto(x, y)</code>	Move turtle to coordinates defined by <code>x</code> and <code>y</code> ; if pen is down, draw line
<code>t.setx(x)</code>	Set the turtle's first coordinate to <code>x</code>
<code>t.sety(y)</code>	Set the turtle's second coordinate to <code>y</code>
<code>t.setheading(angle)</code>	Set orientation of turtle to <code>angle</code> , given in degrees; Angle 0 means east, 90 is north, and so on
<code>t.circle(radius)</code>	Draw a circle with given <code>radius</code> ; the center of the circle is <code>radius</code> pixels to the left of the turtle
<code>t.circle(radius, angle)</code>	Draw only the part the circle (see above) corresponding to <code>angle</code>
<code>t.dot(diameter, color)</code>	Draw a dot with given <code>diameter</code> and <code>color</code>
<code>t.penup()</code>	Pull pen up; no drawing when moving
<code>t.pendown()</code>	Put pen down; drawing when moving
<code>t.pensize(width)</code>	Set the pen line thickness to <code>width</code>
<code>t.pencolor(color)</code>	Set the pen color to <code>color</code> described by string <code>color</code>

# Useful functions in Turtle Class

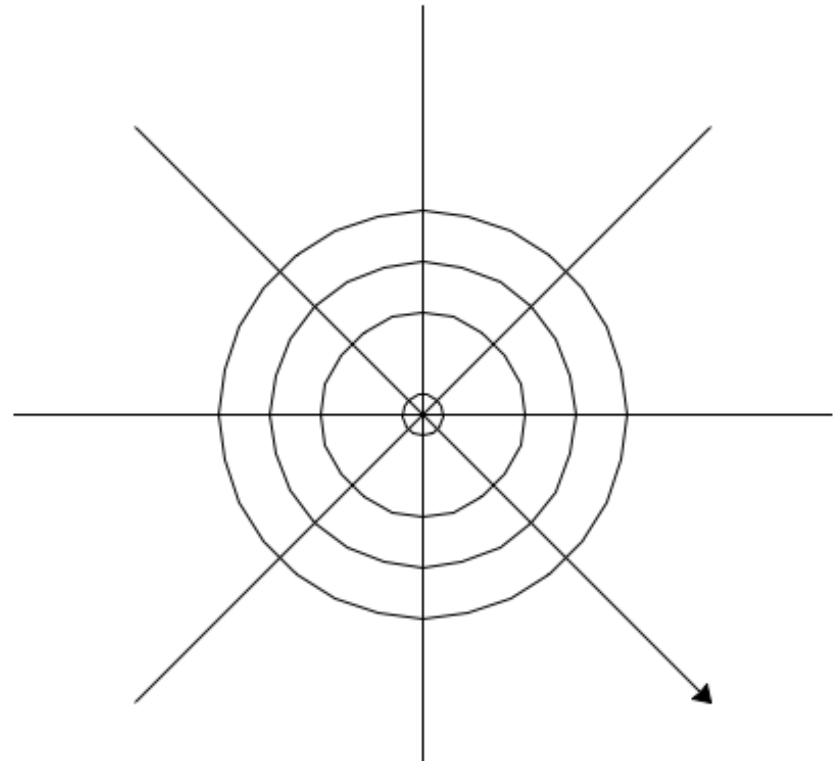
Case Study CS.2 Turtle Graphics 445

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

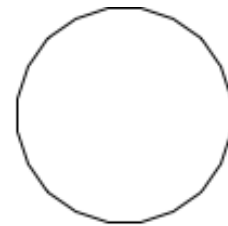
# Exercise 1

- 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 may 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!!**