

ITI 1120

Lab # 1

**An Introduction to the Lab
Environment**

About TA (Teaching Assistant) ...

- Name:
- E-mail:
- Office hour: TBA

Objectives

- Getting familiar with the lab environment
 - Logging In and Logging Out
 - Notable Features
 - E-Mail
 - Blackboard Learn
- Creating and Submitting a Practice Assignment 0
- Your first python program
- Two exercises
- Bonus exercise

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 BlackBoard Learn

- What is at BlackBoard Learn?
 - All the course material is there (think of it as the course web page)
 - Downloading and submitting assignments
 - Announcements
 - Consulting your grades.
 - Discussion group
 - Students are required to check the Blackboard Learn and their OttawaU email frequently for announcements and all material.
- To access BlackBoard Learn:
 - Go to <https://uottawa.blackboard.com/>
 - Enter your InfoWeb user name and password and Sign In.
 - Click on ITI1120 to access the course

Starting Lab 1

- Open a browser and log into Blackboard Learn
- 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](#)
- Then type $1+2*3$, press enter and observe the output
- Then type $(1+2)*3$, 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**

Submit Assignment 0

- Practice **Assignment 0**:
 - Open the file provided with this lab called [UsingIDLE-first-program.pdf](#)
 - Follow the 10 steps to create Python program `a0_XXXXXX.py` and place it in the folder you created called `A0_XXXXXX`.
(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_XXXXXX` folder and then select “Add to A0_XXXXXX.zip”.
 - This will create a zip file with the contents of the directory.
- Go to BlackBoard Learn
 - Find and open Assignment 0
 - Upload your zip assignment file.
 - **ATTENTION**: the assignment has not yet been submitted.
 - Click on the button “Submit”.
 - You will see a confirmation that the assignment has been submitted.

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**

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 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

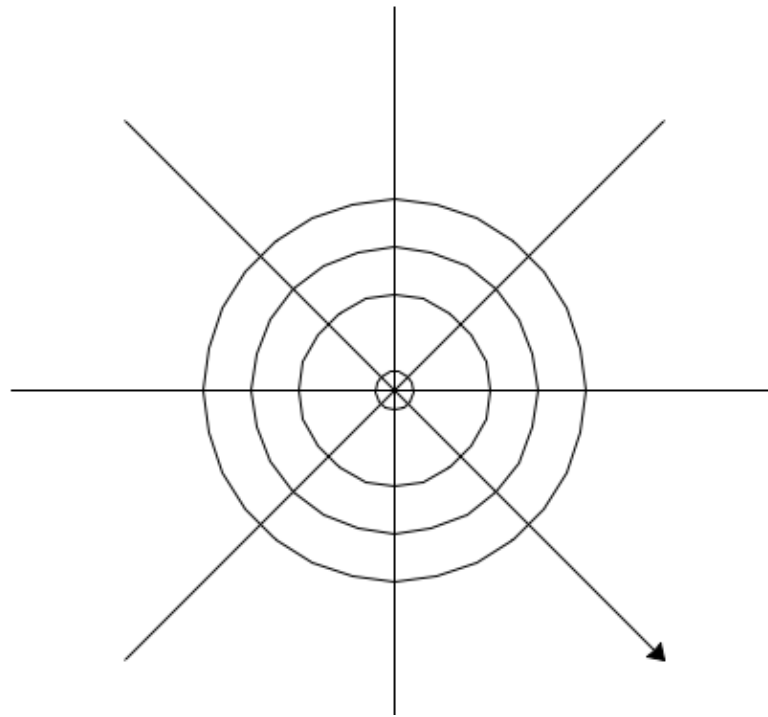
Case Study CS.2 Turtle Graphics 445

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)`

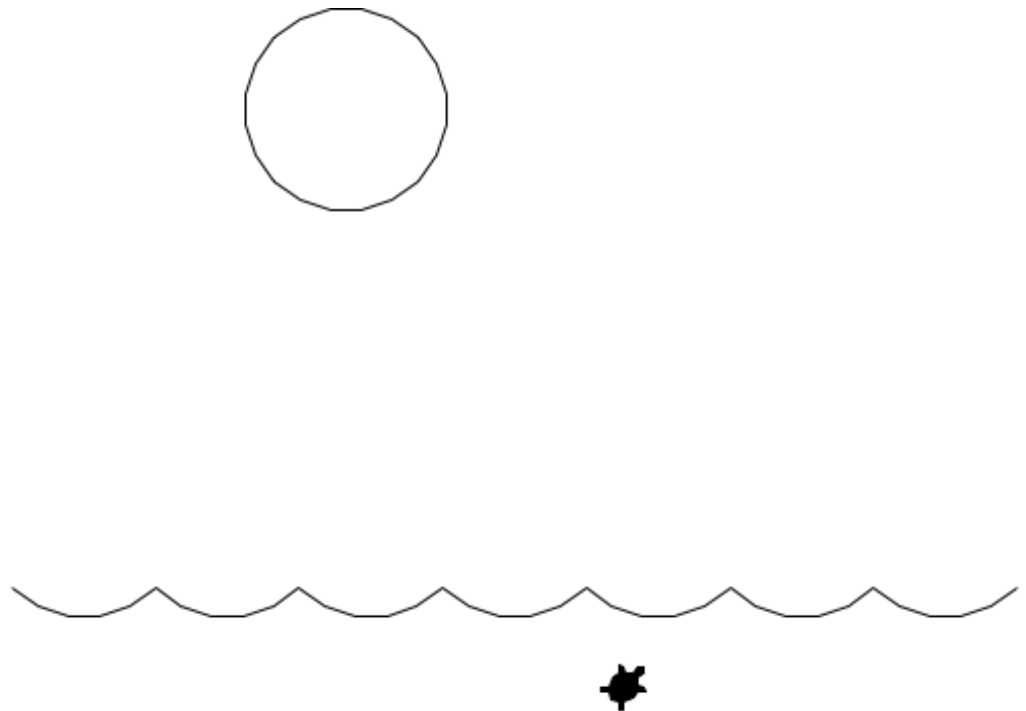
Exercise 1

- 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). **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). Place your code where indicated in file `turtle-ex2.py`



Having fun

Bonus: 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.

Bonus Exercise

- 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 6!!**