RIT-Dubai Spring 2024

GCIS-123

Class Activity & Problem Solving #01

Turtle – Drawing pattern

Due date: February 8, 11:59pm

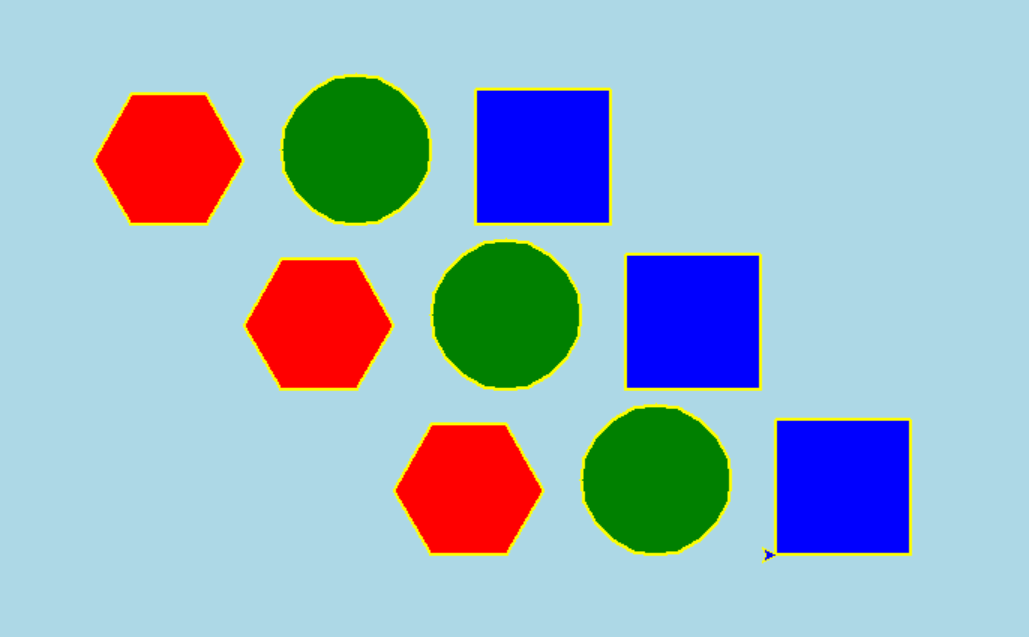
## **Introduction**

Turtle graphics comes with your Python compiler and is a graphical way to learn programming. In order to work with Turtle graphics, you need to import the turtle library. From there you can make the turtle move on your screen.

## **Goals of the Assignment**

This activity involves writing functions that use Python Turtle commands to draw pictures and requires an understanding of fundamental program mechanisms, including Program Behaviour.

Based on the material presented in class, write a python program that produces the below diagram.



## **Before Starting**

Create a new python module in a file named: **yourGroupNumber-activity1.py** and add to this module the logic in the following section (replace *yourGroupNumber* by your correct group).

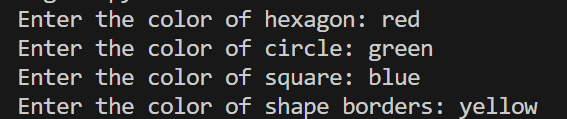
Your program should contain (at least) four functions:

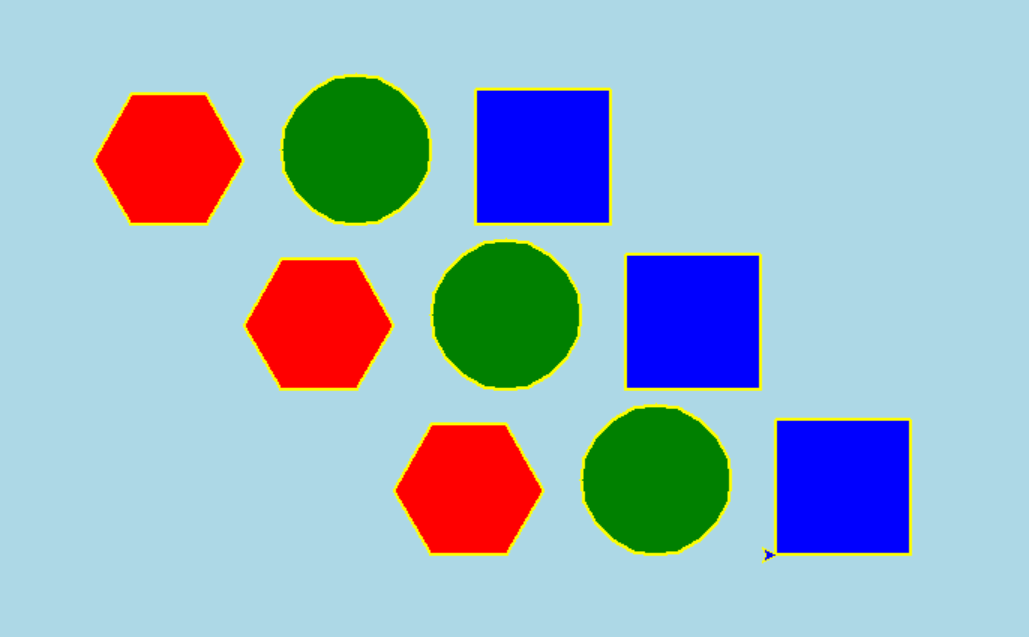
1. ***setPos(turta, x, y)***: the function should accept a turtle object *turta* and the coordinates *x* and *y* (variables). This function sets the position of the turtle to the coordinates *x* and *y* without leaving any trace.
2. ***hexagon(turta, hexa\_color, border\_color)***: the function should accept a turtle object *turta* that draws a hexagon filled with *hexa\_color* color and with a border of *border\_color* color. Each side is of 50 units.
3. ***square(turta, square\_color, border\_color)***: the function should accept a turtle object *turta* that draws a square filled with *square\_color* color and with a *border\_color* border. Each side is of 90 units.
4. ***circle(turta, circle\_color, border\_color)***: the function should accept a turtle object *turta* that draws a circle filled with *circle\_color* color and with a *border\_color* border.
5. ***pattern(turta, hexa\_color, circle\_color, square\_color, border\_color)***: the function should accept a turtle object *turta* that draws a hexagon filled with *hexa\_color* color first followed by a circle filled with *circle\_color* then followed by a square filled with *square\_color*.
6. Based on what you have learned in this course and using the appropriate functions defined in above, you are asked to implement a python program (i.e. ***main()***) that draws the below diagram (see Run#1 and Run #2).

Please be aware that your program should commence by prompting the user to input the colors for the shapes and subsequently, the drawing window (canvas) will open, initiating the turtle's drawing process.

***Sample Outputs:***

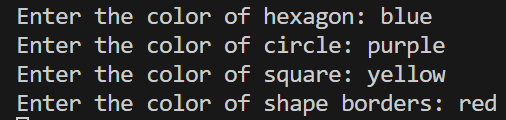
***Run #1:***

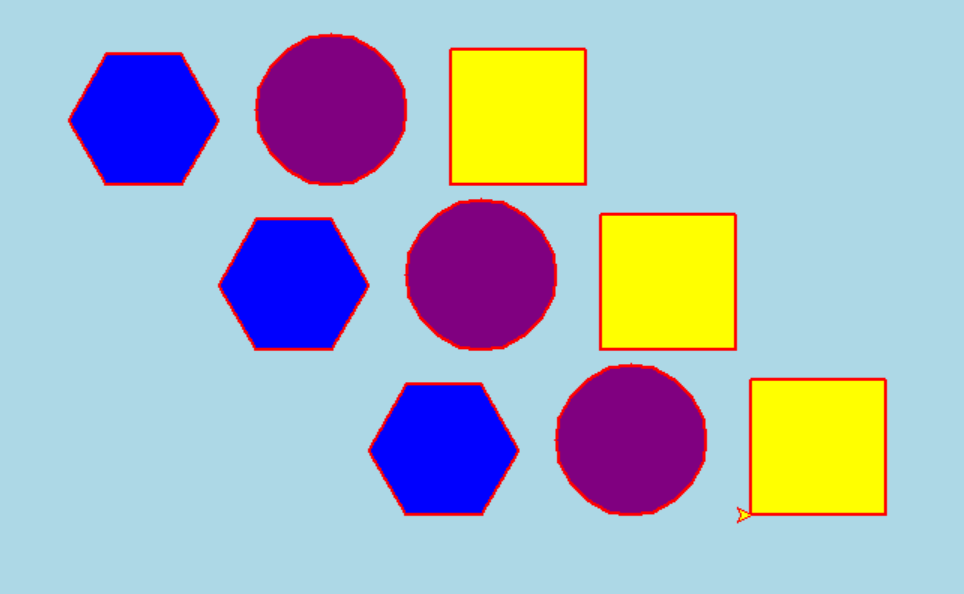




***Sample Outputs:***

***Run #2:***

******

******

## **Rubrics**

|  |  |
| --- | --- |
| Create the file yourGroupNumber-activity1.py | 2 points |
| Write docstrings for the file including the manifesto | 8 points |
| Asks parameters to the user to customize the drawings.  Style and Documentation (best practice variable naming as son on) | 5 points |
| Properly passes arguments for drawing hexagon, circle, square, pattern and so on functions | 10 points |
| Implements functions for drawing the diagram elements with parameters | 15 points |
| Draws a diagram with at least three rows including the shapes in order as requested | 40 points |
| Drawing fits within the canvas window without cutoff. | 5 points |
| Fills shapes with colors, and border lines match fill colors. | 5 points |
| Displays the drawing for the user to view after the user enter the requested value and before exiting the program. | 5 points |

## **Submission Instructions**

* + - 1. Be sure to replace yourGroupNumber-activity1.py by your group number (e.g., for group 1, your module’s name will be *group1-activity1.py*).
      2. You are not allowed to use any concepts not explained in class this term (e.g., for, while, etc.). Otherwise, -50% will be applied.
      3. For late submission, 10% of the work’s assigned mark will be deducted, for every calendar day without exception, for a maximum of 2 days, after which a zero will be given on the missed assessment (Please refer to the syllabus).
      4. Include the appropriate internal-documentation (i.e. comments & docstring) for each function you implemented in this work.
      5. Include the docstring for your module (i.e. yourGroupNumber-activity1.py). This docString should include first, the names of all the members of your group, Second, the contribution of each member to this activity and then the description of your module briefly. Otherwise -8% will be applied (see the rubric above).
      6. Upload **ONLY** **the required file** to the MyCourses Assignment box as (yourGroupNumber-activity1.zip). Otherwise -3 % will be applied (only one team-member needs to submit to MyCourses).
      7. Be sure that you have pushed **ALL the file** to your GitHub classroom repository (**for each group)**. All the members should commit to this repository- at least two commits per student. Otherwise -10% will be applied. DON’T WAIT UNTIL THE LAST DAY TO CHECK YOUR GITHUB AND FIX YOUR PROBLEMS.
      8. **Reminder:** Note regarding group work/group projects: all students in a group are considered a single unit and will be held accountable for any offense perpetrated by the group as a whole or an individual member of that group. Note that there should be no distinction in cheating incidents between the student that provided the material and the student that received the material. Both must receive the same consequences.