

# Basics of Programming

## L01

Apr 02, 2019

Dr. Ram P Rustagi  
Professor, CSE Dept  
KRP, KSGI  
[rprustagi@ksit.edu.in](mailto:rprustagi@ksit.edu.in)

# Overview

- Basics of programming
- Need for
  - variables
  - variable types
  - loop
  - function definition
- L02: Writing correct programs

# Resources and Acknowledgements

- Intro to Programming with C++
  - Abhiram Ranade, Prof CSE, IIT Bombay
- A first course in programming
  - <https://introcs.cs.princeton.edu/python/home/>
  - <https://introcs.cs.princeton.edu/java/home/>
- Python for everybody
  - <https://www.py4e.com>
- Web Applications for everybody
  - <https://www.wa4e.com>
- Turtle Graphics
  - <https://docs.python.org/3/library/turtle.html>

# What is Programming

- A creative activity by CS Engineer
- An engineer's pride
  - build thing and see it working/used
- A CS engineer pride
  - Create programs and see it getting used
    - Meets (exceeds) given requirements
      - Anticipate future requirements and make provisions
    - No crashes
    - No side effects
    - Scope for enhancement
- Goal: Learn how to make computers do things we need

# What is a Program?

- Program
  - A precise description of the calculations we want the computer to perform
  - By feeding different programs to a computer you can make it do different calculations.
- This course
  - Introduces you to tenets of programming
- Programming language
  - A language that is used to write programs
    - Java, python, C/C++, SQL, ...

# Starters

- Work with pictures (polygons)
- Turtle
  - Moves on the screen
  - It has a pen, so it draws on the screen
    - pen should be down
  - When pen is up, it just moves, but does not draw
- Drawing pictures is fun based learning
  - Mastering of pictures drawing will make you learn lot of basics of programming
- Basic APIs:
  - `forward()` , `left()` , `right()` ,
  - `penup()` , `pendown()`
  - `wait()`

# Turtle: Drawing a Square

```
from turtle import *  
# Draw a square of each side 100 pixels  
forward(100)  
left(90)  
forward(100)  
left(90)  
forward(100)  
left(90)  
forward(100)  
left(90)
```

**Q: How to make square of each side of 150 pixels?**  
**How many changes we need to make?**

# Drawing a Square: Use Variable

```
from turtle import *  
# Draw a square whose length can be changed.  
x=150  
forward(x)  
left(90)  
forward(x)  
left(90)  
forward(x)  
left(90)  
forward(x)  
left(90)
```

**Q: How to draw a pentagon, hexagon or a general polygon**



# Drawing a Pentagon

```
from turtle import *  
# Draw a pentagon whose length can be changed.  
x=150  
forward(x) ; left(72)  
forward(x) ; left(72)  
forward(x) ; left(72)  
forward(x) ; left(72)  
forward(x) ; left(72)
```

**Q: How to draw a hexagon?**

# Drawing a Hexagon

```
from turtle import *  
# Draw a Hexagon  
x=150  
forward(x) ; left(60)  
forward(x) ; left(60)  
forward(x) ; left(60)  
forward(x) ; left(60)  
forward(x) ; left(60)
```

**Q: How to draw a polygon of  $n$  sides**

**Do you need to repeat writing  $n$  lines**

**Can program repeat these using some kind of loop**

# Drawing a Octagon: Using loops

```
from turtle import *
x=150
n=8
angle = 360//n # integer division
# repeat the set of lines n times.
for i in [1,2,3,4,5,6,7,8]:
    forward(x); left(angle)
# alternatively
for i in range(n):
    forward(x); left(angle)
```

**Q: Can we draw a heptagon?**

# Drawing a Octagon: Using loops

```
from turtle import *
x=150
n=7
angle = 360//n # integer division
# repeat the set of lines n times.
for i in range(n):
    forward(x); left(angle)
```

**Q:What happens? Do you notice a gap? why?**

- We need real number value for angle.
  - 360 is not perfectly divisible by 7.
- Replace `angle = 360/7` #real number division

# Working with Variables

- Integer - Represented as 2's compliments

`int`

python: No limit on range

java:  $-2^{31}$  to  $2^{31}-1$

- Real numbers:

python: `float`

java: `double`, `float`

- Strings: array of letters

- Character

– Not in python

- Boolean

`True` or `False`

# Integer Arithmetic

- Consider  $a = bq + r$
- As long as  $a$  and  $b$  are non-negative integer
  - $q$  and  $r$  are non-negative integers
- What happens when  $a$  is -ve and  $b$  is +ve
  - Language implementation varies
  - Remainder: java sign of dividend, python: sign of divisor
- Examples (integer division)
  - $-42/5, -42\%5, 42/-5, 42\%-5$
  - java:  $-8, -2, -8, 2$
  - python:  $-9, 3, -9, -3$

# Back to Polygons

- Draw 2 consecutive squares

```
from turtle import *
x=100
n=4
angle = 360//n # integer division
# repeat the set of lines n times.
for i in range(n):
    forward(x); left(angle)
penup()
forward(x)
pendown()
for i in range(n):
    forward(x); left(angle)
```

# Draw 3 Consecutive Squares

```
from turtle import *
x=100; n=4
angle = 360//n # integer division
# repeat the code 3 times
for i in range(n):
    forward(x); left(angle)
penup(); forward(x); pendown()
for i in range(n):
    forward(x); left(angle)
penup(); forward(x); pendown()
for i in range(n):
    forward(x); left(angle)
penup(); forward(x); pendown()
```



# Need of Function

- Define a function to be reused
  - avoids rewriting the code again and again.

```
from turtle import *
x=100; n=4
angle = 360//n # integer division
# define a function
def polygon(n,x):
    for i in range(n):
        forward(x); left(angle)
    penup(); forward(x); pendown()
#invoke the function
polygon(4, 50)
polygon(5, 60)
polygon(4, 50)
```

# Black Square

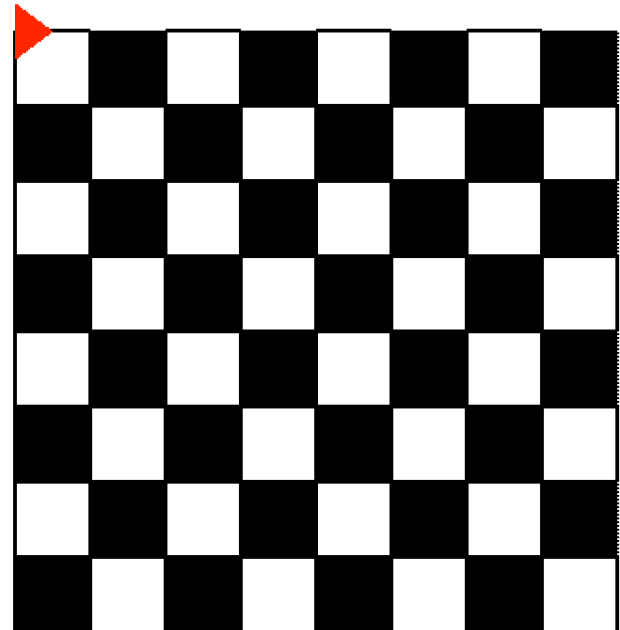
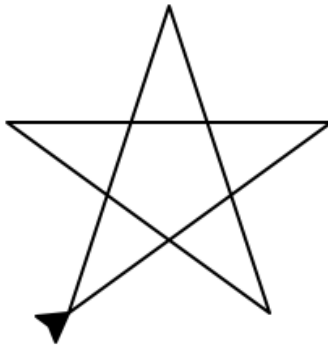
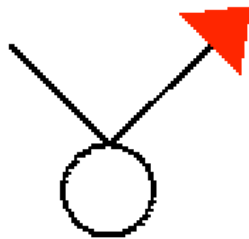
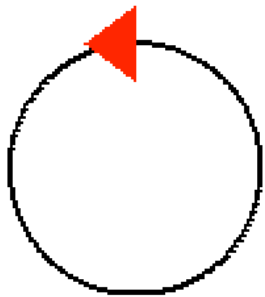
```
from turtle import *

def blacksquare(x):
    angle=90
    for i in range(x//2):
        forward(x); left(90)
        forward(1); left(90)
        forward(x); right(90)
        forward(1); right(90)

blacksquare(100)
```

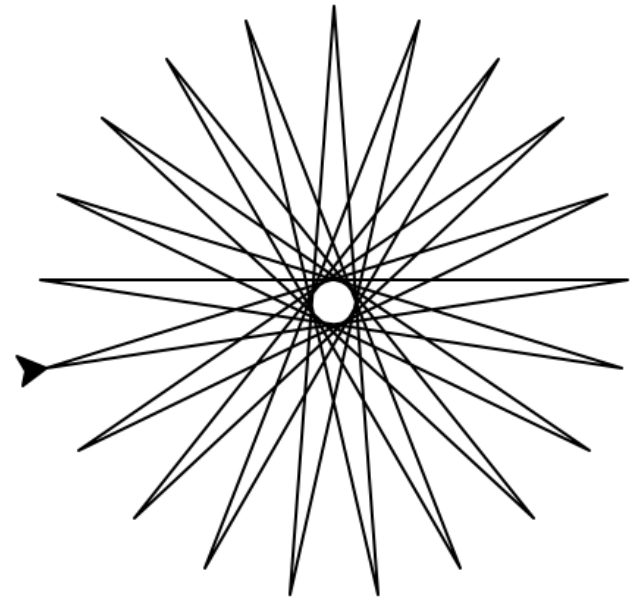
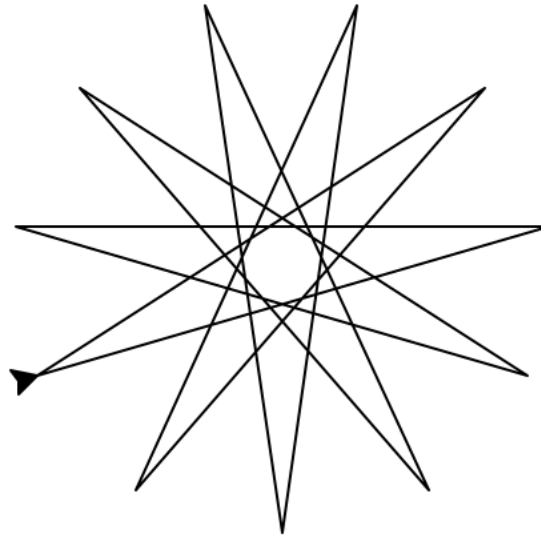
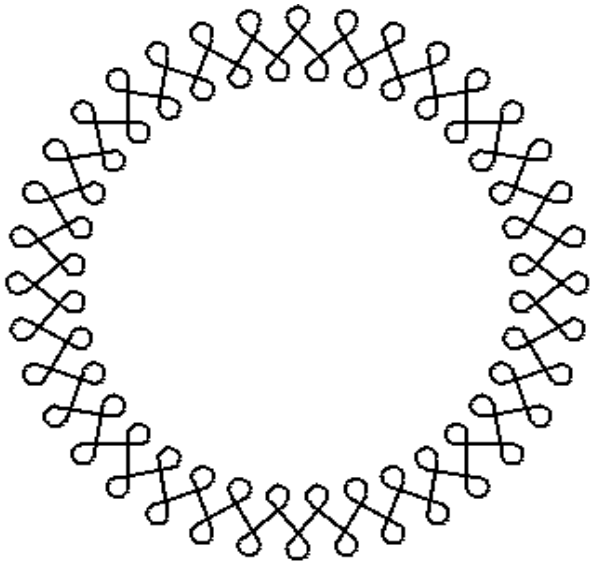
# Exercises

- Write programs to draw following figures
  - Circle (perimeter=360px)
  - Rectangle(sides=100px, 60px)
  - Badge
  - A 5-pointed star
  - A 7-pointed star
    - How many you such stars you can draw
  - A chessboard



# Advanced Exercises

- Draw following pattern with 36 repetitions
  - ref: Chap 01, Book by Abhiram Ranade
- Draw pointed  $n$ -star polygon e.g.  $n=11$ ,  $n=21$



# Summary

- Introduction of
  - Variables
  - variable types (int, float)
  - loops (for loop)
  - function definition invocation
- Logical thinking for drawing figures

# Questions

