

CS 101

Computer Programming and Utilization

Instructor: Rohit Gurjar

All material based on Prof. Abhiram Ranade's book and slides
(and previous course instructors)

Getting started with programming

The C++ Programming Language

- Designed by Bjarne Stroustrup, 1980s.
- Evolved out of the C programming language.
- C++ is a powerful, complex language.
- We will not study all of it.
- We will lay the foundation for learning advanced features later.

The Programming Environment

- Initial weeks: C++ augmented with Simplecpp
- Simplecpp is a C++ library developed in IITB by Prof. Abhiram Ranade
 - Provides facilities convenient to learners
 1. Simplified syntax
 2. Graphics programming – more fun!
 - Download from www.cse.iitb.ac.in/~ranade/simplecpp
Available as Linux/Mac OS library or as IDE for windows and Linux
- Later weeks: Only C++
 - We may continue to use Simplecpp graphics

Let us write some simple C++ programs

- The programs will draw pictures on the screen.
- Use “Turtle Simulator” contained in simplecpp
 - Based on Logo: A language invented for teaching programming to children by Seymour Pappert et al.
 - We “drive” a “turtle” on the screen!
 - To drive the turtle you write a C++ program.
 - Turtle has a pen, so it draws as it moves.

Drawing pictures seems too much fun?

“You master picture drawing, you master programming!”

The first program

```
#include <simplecpp>
main_program{
    turtleSim();
    forward(100);    right(90);
    forward(100);    right(90);
    forward(100);    right(90);
    forward(100);
    wait(5);
}
```

- “Use simplecpp facilities”
- Main program begins
- Start turtle simulator
 - Creates window + turtle at center, facing right
- **forward(n) :**
 - Move the turtle n pixels in the direction it is currently facing.
- **right(d) :**
 - Make turtle turn d degrees to the right.
- **wait(t) :**
 - Do nothing for t seconds.

How to run the program

- First install `simplecpp` on your computer. TAs will help with this in the first week.
1. Type the program in a file, using a text editor.
 - For example, vim (from terminal), gedit (on linux), TextEdit (on mac)
 2. Open Terminal, go to the folder where you have created the file (can use `cd` command, or open the folder in terminal using right-click).
 3. Compile the program by typing the following on Terminal, and pressing enter / return
`g++ square.cpp`
 4. Execute / run the program by typing the following on Terminal, and pressing enter / return
`./a.out`

Exercises

- Write a program to draw
 - A smaller square
 - A rectangle
 - An equilateral triangle
 - A pentagon
- Remember that the external angles of a polygon add up to 360 degrees.

Repetition in code — use **repeat**

```
#include <simplecpp>
main_program{
    turtleSim();
    forward(100); right(90);
    forward(100); right(90);
    forward(100); right(90);
    forward(100);
    wait(5);
}
```

```
#include <simplecpp>
main_program{
    turtleSim();
    repeat(4){
        forward(10);
        right(90);
    }
    wait(5);
}
```

- Both the programs draw a square.
- But, the final direction of the turtle is different.

Use of repeat statement

```
repeat (n) {
```

```
  ⋮
```

```
  body
```

```
  ⋮
```

```
}
```

← **n** must be a positive number

← **body** is one or more statements

- What will it do?
 - **body** is executed **n** times.
- This is a “**loop**”. There are other kinds of loop, which we will see later.
- Each execution of the **body** is called an “**iteration**”.

How to draw a polygon

```
#include <simplecpp>
main_program{
    turtleSim();
    cout << "How many sides?";
    int nsides;
    cin >> nsides;
    repeat(nsides){
        forward(100);
        right(360.0/nsides);
    }
    wait(10);
}
```

```
cout << msg;
```

- Print message `msg` on the screen.

```
int nsides;
```

- Reserve a cell in memory in which I will store some integer value, and call that cell `nsides`.
- You choose the name as you wish, almost!

- `int` : abbreviation of "integer"

```
cin >> nsides;
```

- Read a value from the keyboard and put it in the cell `nsides`.

```
repeat(nsides)
```

- repeat as many times as content of `nsides`

```
360.0/nsides
```

- result of dividing 360 by content of `nsides`.

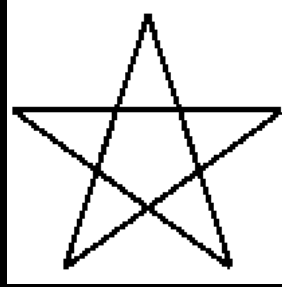
More simplecpp commands

- `left(A)` : turn left A degrees. Equivalent to `right(-A)`
- `penUp()` , `penDown()`: Causes the pen to be raised, lowered
 - Drawing happens only if the turtle moves while the pen is low.
- `hide()` : hide the turtle (triangle).
- `sqrt(x)` : square root of x.
- `sine(x)` , `cosine(x)` , `tangent(x)` : x should be in degrees.
- `sin(x)` , `cos(x)` , `tan(x)` : x should be in radians.
- Also commands for `arcsine` , `arccosine` , `arctangent`... See book.

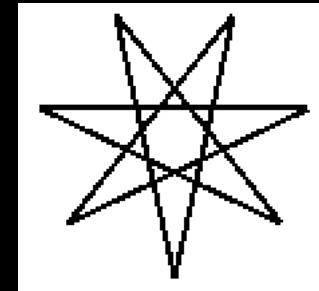
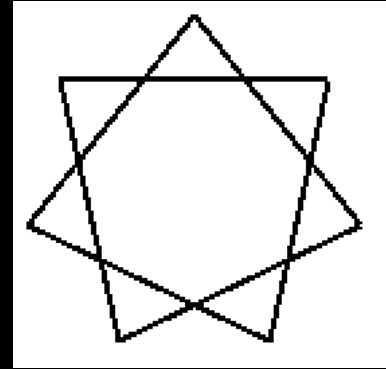
What we have learnt so far

- `Repeat()` causes repeated executions of some statements.
- `cin` and `cout` statements can be used to read from keyboard and to type messages to the screen.
- We have commands to compute mathematical functions as well as lift the pen up and down.

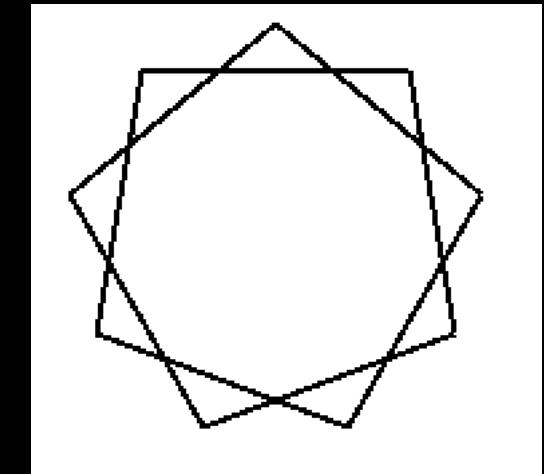
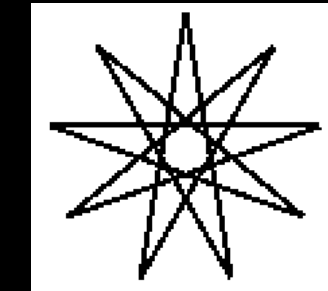
Exercise: Stars



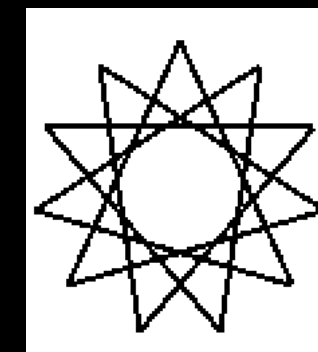
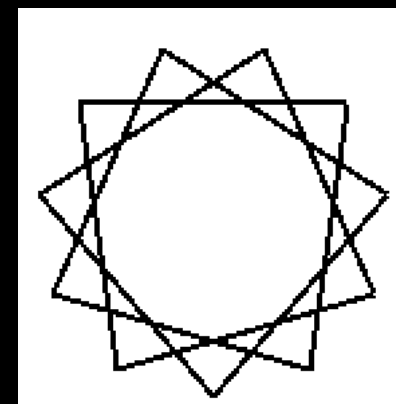
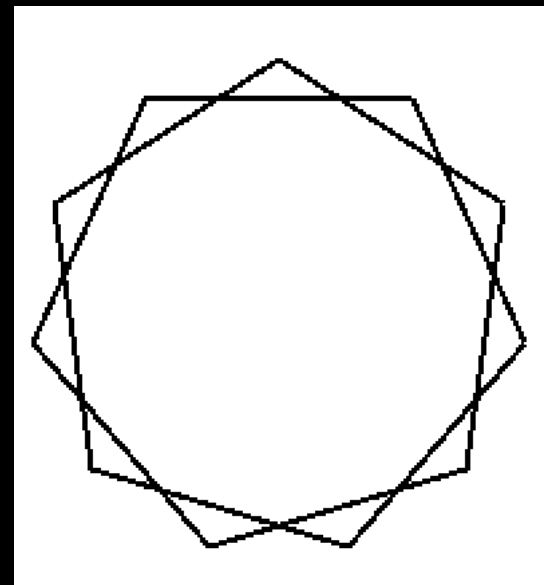
5-star



7-stars



9-stars



11-stars

Exercises

