#### CS 101: Computer Programming

Jan-Apr 2017

Sunita Sarawagi (cs101@cse.iitb.ac.in)

Course webpage: http://www.cse.iitb.ac.in/~cs101/

Lecture 1: Introduction

#### **About These Slides**

- Based on Chapter 1 of the book
   An Introduction to Programming Through C++ by Abhiram Ranade (Tata McGraw Hill, 2014)
- Original slides by Abhiram Ranade
  - First update by Varsha Apte
  - Second update by Uday Khedker

#### Computers are everywhere

- So far almost all of you have <u>used</u> a computer
  - Desktop
  - Laptop
  - <u>Smartphone</u>(is a computer)







#### A Computer Can Do Many Things



#### Control railway switching and signals



#### Recommendations of what to buy



#### Weather prediction



Perform super-realistic graphics and movement



# Computer: a programmable machine



A Program controls what calculations to perform

# A computer is

a giant electrical circuit that can

- Receive data from the external world
  - data = numbers,
  - images, sounds can also be represented using numbers and hence fed to a computer
- Perform calculations on the data it receives
- Send the results back to the external world
- What calculations to perform: determined by a program that must be loaded in the computer

## Programs

- 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 tells you how to construct ("write") programs.
- Special notation is to be used to write programs: "Programming Language"

#### Programming the turtle to draw a square

Instructions the turtle understands

- penUp()
  - Will not draw while moving
- penDown()
  - Will draw while moving
- forward (x): Move forward x pixels
  - E.g. forward(50) moves the turtle forward 50 pixels
- right (x): turn right by x degrees
- left(x): turn left by x degrees



# Programming a turtle to draw a square (continued)

- With these instructions, make the turtle move in such a way that we will draw a square of side length 200
- Note: by default, in the beginning, the turtle faces towards east, and the pen is down

# Our programming platform

• We will use Prutor (Programming Tutor) System

https://cs101.cse.iitb.ac.in

We use the same system in the class

 We will learn a more general and more powerful approach of running programs from command line, later in the course

#### The C++ Programming Language

- Designed by Bjarne Stroustrup, 1980s
- Derived from the C programming language
- Substantial evolution (still continues)
- Early part of our course: C++ augmented with a package called simplecpp

(designed by Abhiram Ranade)

More fun and easier to use than bare C++ Builtin graphics

#### A simplecpp program



## **General Ideas**

#include<simplecpp>

main\_program{

turtleSim();

forward(200); right(90); forward(200); right(90); forward(200); right(90); forward(200); Commands or statements terminated by semicolon ";"

This sequence of commands in C++ is the program

Some commands need additional information called *arguments* 

- 90 is the argument to the command right
- 200 is the argument to the command forward

## General Ideas (contd)

#include<simplecpp> main program{ turtleSim(); forward(200); right(90); forward(200); right(90); forward(200); right(90); forward(200);

Commands are generally executed from top to bottom, left to right. (we can override this default)

## General Ideas (contd)

- Compiling a program: Translating it into a form that your computer can understand
- The result of compilation: An *executable* file
- This is done internally by Prutor (By invoking a C++ compiler)

#### How to Draw An Octagon?

#include <simplecpp> main program{ turtleSim(); forward(100); right(45); forward(100); right(45);

- Commands seem
   quite repetitive?
- There's a better way!

## A Better Way

#include <simplecpp> main\_program{ turtleSim(); repeat(8){ forward(100); right(45);

repeat statement: repeat (n) { some commands is a command that can be compiled by simplecpp The instructions within {...} are repeated *n* times Each round of execution is called an *iteration* 

#### How to Draw a Polygon

- We have removed repeated occurrences of a command
- Can we generalize it further to draw a polygon of any number of sides??
- Yes! By using variables!

#include <simplecpp> main\_program{ turtleSim(); cout << "No. of sides?"; int noofsides; cin >> noofsides; repeat(noofsides){ forward(10); right(360.0/noofsides);

## Explanation

#include <simplecpp> main program{ turtleSim(); cout << "No. of sides?" int noofsides; cin >> noofsides; repeat(noofsides) { forward(200); right(360.0/noofsides); Print the sentence within the quotes on the screen (required in command line, not in Prutor)

Tell the computer: Reserve space in your memory where I can store an integer (int). I will refer to it by the name noofsides

Read the number that the user types and store it into the space in memory named noofsides

Use the integer stored in the space in memory which is named noofsides

Divide the number 360 by the number stored in the space named noofsides and pass the result as an argument to this command

### More Commands/Functions

- sqrt(x) : square root of x
- Trigonometric functions,
  - x is in radian: sin(x), cos(x), tan(x)
  - x is in degree sine(x), cosine(x), tangent(x)
- Also for arcsine, arccosine, arctangent etc.

# Repeat Statement Within Another Repeat Statement

```
repeat(4){
   repeat(3){
      forward(50); penUp();
      forward(50); penDown();
   right(90);
```

### **Nested Repeat Statements**

Basic rule:

repeat(n){ yyy }

means

Statements yyy to be executed x times

- If yyy contains repeat (m) {zzz},
  - Then the zzz is executed m times in each iteration of outer repeat
  - Thus zzz will get executed n X m times

What will the program fragment on previous slide do?

#### **Nested Repeat Statements**

```
repeat(4){
   repeat(3){
    forward(50); penUp();
    forward(50); penDown();
   right(90);
```

It will draw a square with dashed lines

#### What Does the Following Program Do?

```
#include <simplecpp>
main_program{
    cout << "a";
    repeat(5){
         cout << "b";
         repeat(2){ cout << "c"; }
         cout << "d";
```

#### Answer

#### The program prints

abccdbccdbccdbccdbccd

## Remarks: Some Terms

#### Control is at statement w

The computer is currently executing statement w

#### Control flow

The order in which statements get executed.

- Execution starts at top and goes down
- Retraced if there is a repeat statement
- Later we will see selective executuion
- Variable: used for storing data
  - Computer memory: blackboard
  - Variable: Space on the board in which a value can be written
  - Variables have names, e.g. noofsides. We can use the name to refer to the value written in the variable.

(Sequence) (Iteration) (Selection)

# The Spirit of The Course

Learn C++ statements/concepts

We have covered a lot of ground in this lecture, even if it doesn't seem so

- Learn how to express problems you want to solve using C++.
- Goal: if you can solve a problem by hand, possibly taking an enormous amount of time, by the end of the course, you should be able to write a program for it

Learn new ways of solving problems!

#### How to master the course

- Do not be afraid of using the computer
- "What if I write xyz in my program instead of pqr?"
  - Just do so and find out
- Be adventurous.
- Exercise your knowledge by writing programs that is the real test

# Why Picture Drawing?

- Picture drawing requires calculation e.g. 360.0/noofsides
- "Draw a triangle of sides with lengths 3, 4, 5 units"

You will need to do trigonometric calculations to find out the angles between the sides

 More interesting calculations will be needed to draw more interesting drawings

# Why Picture Drawing (contd)

- Interesting pictures contain patterns
- Most interesting calculations of any kind (not necessarily picture drawing) also contain patterns
- The pattern in the calculations must be mirrored by patterns in program
- Example: if a certain sequence of computations needs to be repeated, then do not repeat it textually, but put it in a repeat statement

A pattern with 36 repetitions. You know enough to write a program to do this! Try it.

