

CS 101: Computer Programming & Utilization

Autumn 2017

Umesh Bellur

(cs101@cse.iitb.ac.in)

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



Acknowledgements

- 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
 - Third update by Sunita Sarawagi

Computers are everywhere

So far almost all of you
have used a computer

- Desktop
- Laptop
- Smartphone
(is a computer)
- Car



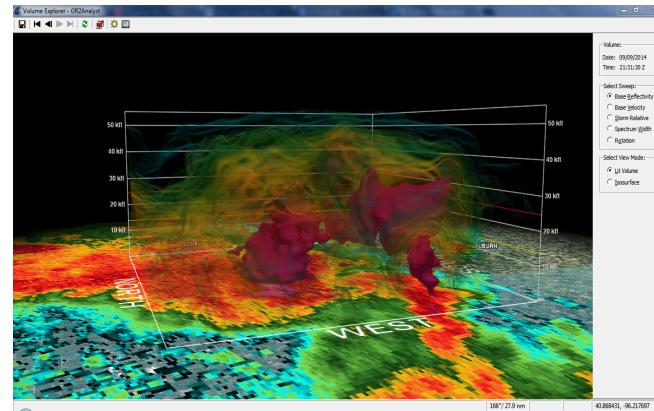
A Computer can do many things



Control railway switching and signals

A screenshot of the Flipkart website. At the top, there are navigation links for 'Sell', 'Download App', 'Flipkart First', and '24x7 Customer Support'. Below that is a search bar with the placeholder 'So, what are you wishing for today?'. The main menu includes categories like 'ELECTRONICS', 'MEN', 'WOMEN', 'BABY & KIDS', 'HOME & KITCHEN', and 'BOOKS & MEDIA'. A 'My Account' section is highlighted in blue. To the right, a 'Recommendations For You' section is shown, featuring two product images: a TP-LINK 150Mbps Wireless N USB Adapter and a Logitech B175 Wireless Mouse (Wireless). Both products have their names, prices, and discount percentages listed below them.

Recommendations of what to buy



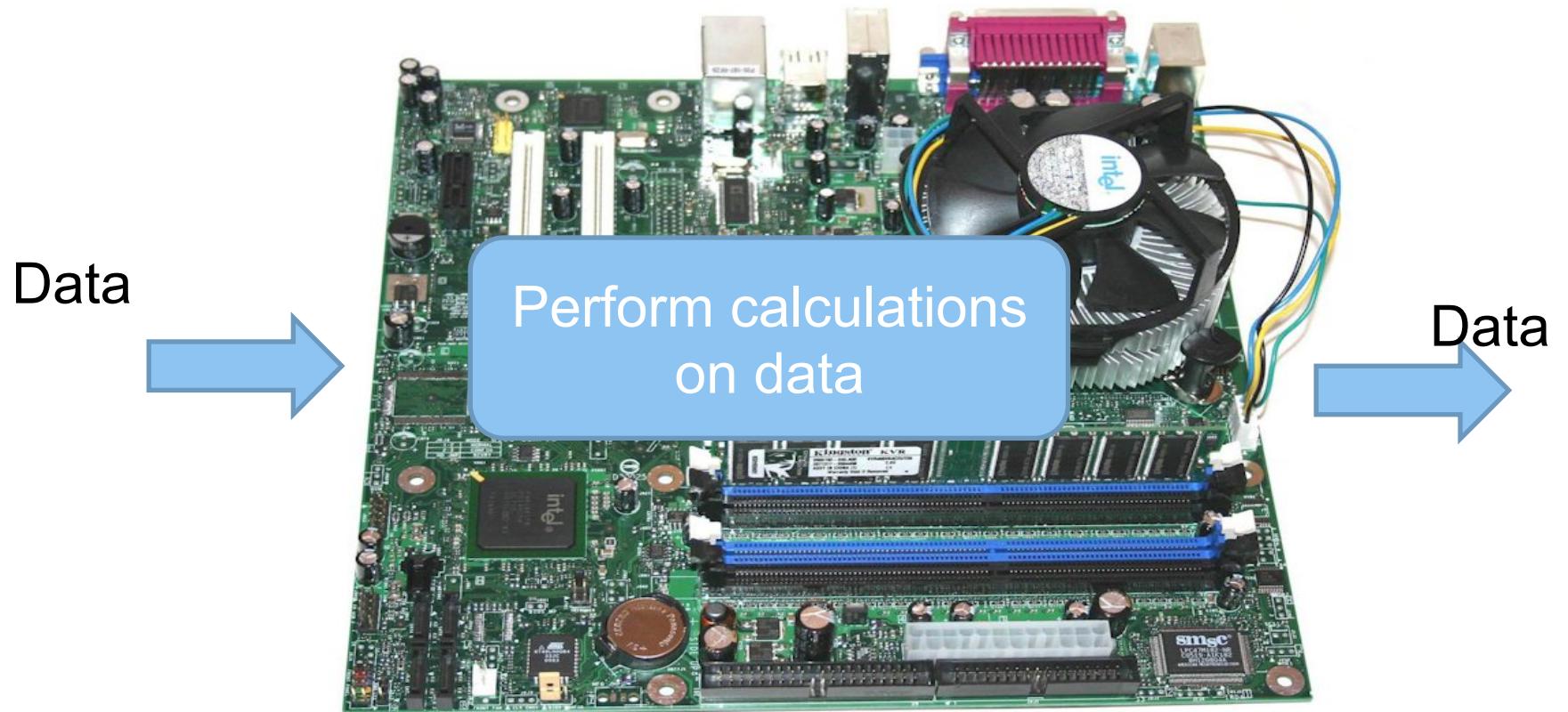
Weather prediction



Perform super-realistic graphics and movement

How ?

Computer: a programmable machine



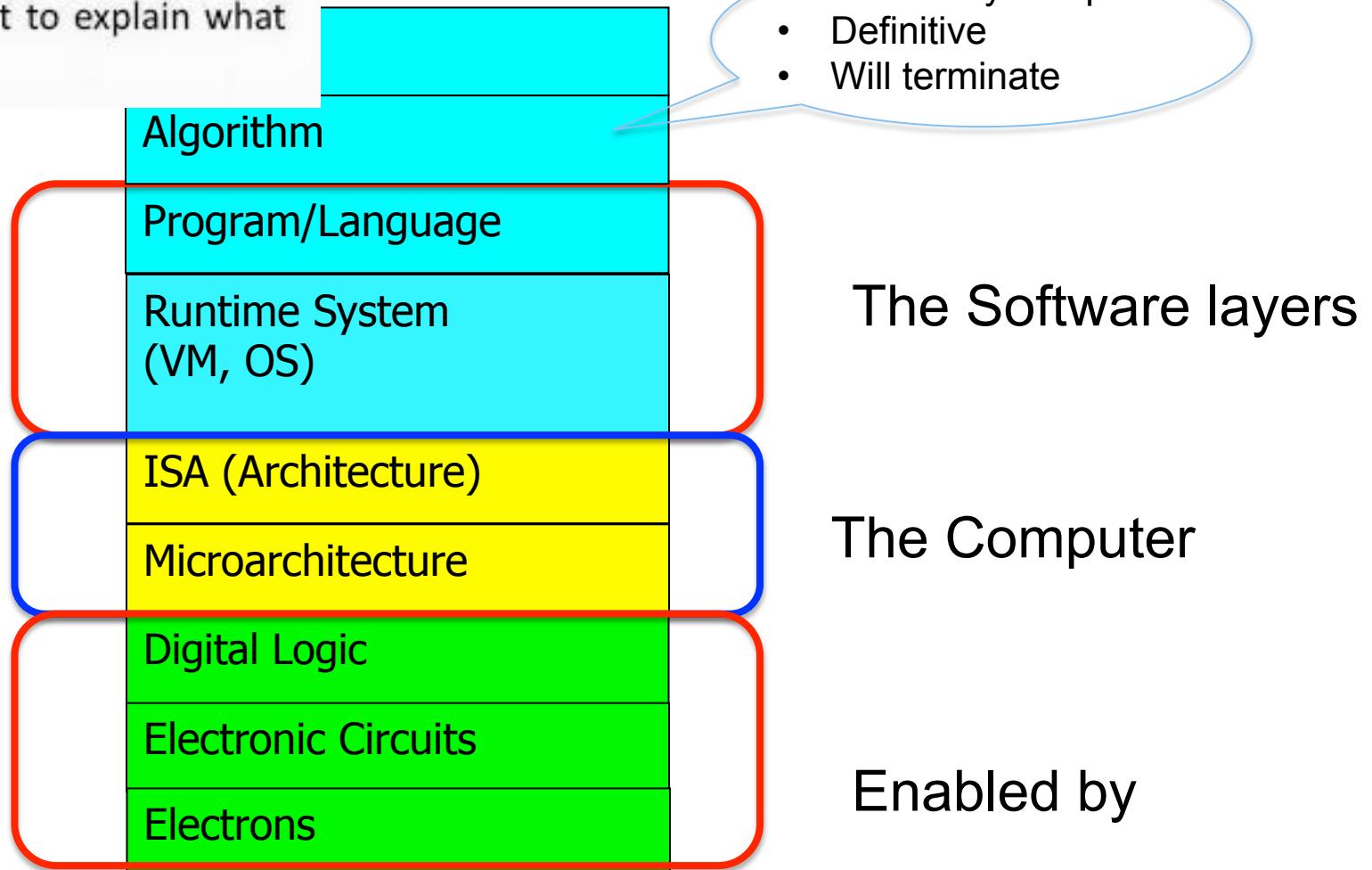
A PROGRAM controls what calculations to perform

algorithm

noun

Word used by programmers when they do not want to explain what they did.

Definitions and Abstraction



The Power of Abstraction

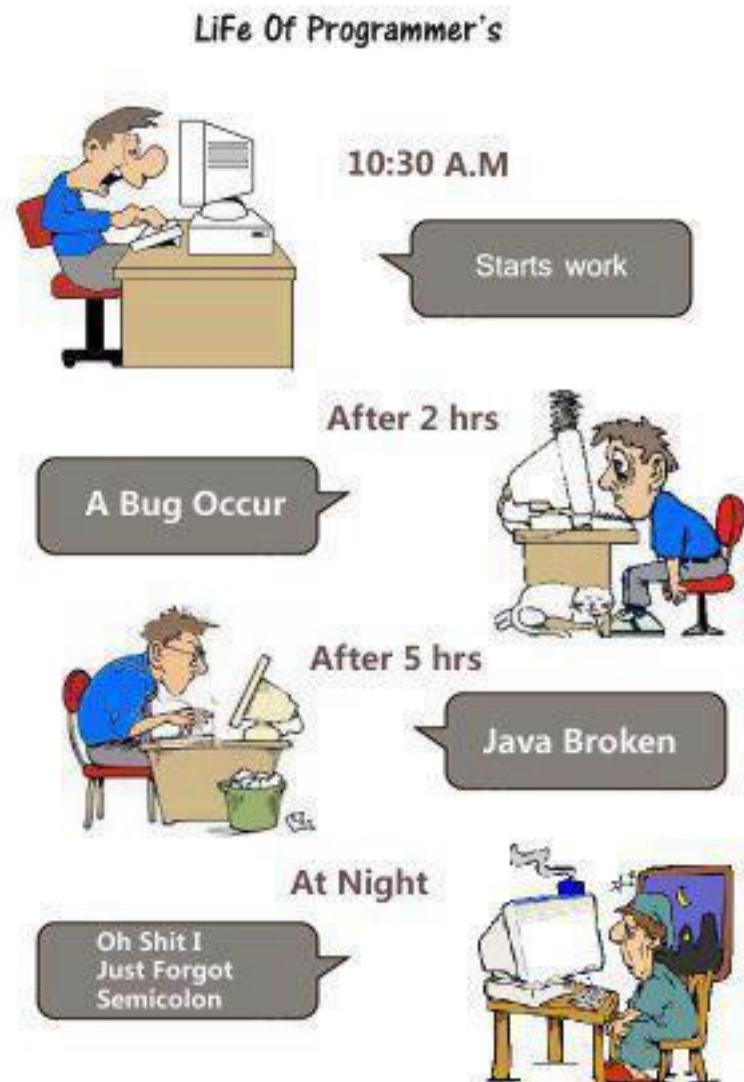
- **Levels of transformation create abstractions**
 - Abstraction: A higher level only needs to know about the interface to the lower level, not how the lower level is implemented
 - E.g., high-level language programmer does not really need to know what the ISA is and how a computer executes instructions
- **Abstraction improves productivity**
 - No need to worry about decisions made in underlying levels
 - E.g., programming in Java vs. C vs. assembly vs. binary vs. by specifying control signals of each transistor every cycle
- But its good to know how things work beneath the hood.
 - Driving a car.

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
 - Special notation is to be used to write programs: “**Programming Language**”
- By feeding different programs to a computer you can make it do different calculations.
- This course tells you how to construct (“write”) programs.



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 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 (that is continuing).
- Early part of our course: C++ augmented with a package called **simplecpp**
 - More fun and easier to use than bare C++ Built-in graphics

A simplecpp program

```
#include <simplecpp>
main_program {
    turtleSim();
    forward(200);
    right(90);
    forward(200);
    right(90);
    forward(200);
}
}
```

the program will use the simplecpp package.

Your commands within these braces {...}package.

Start the turtle simulator (open a window)

Move forward 200 units

Turn right 90 degrees

Program exits

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);  
    }  
}
```

```
#include <stdio.h>  
int main(void)  
{  
    int count;  
    for(count=1; count<=500; count++)  
        printf("I will not throw paper airplanes in class.\n");  
    return 0;  
}
```

repeat statement:

```
repeat (n) {
```

some commands

```
}
```

is a command that can be
compiled by simplecpp

The instructions within {...} are



ition is

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 numsides;
    cin >> numsides;
    repeat(numsides){
        forward(10);
        right(360.0/numsides);
    }
}
```

Explanation

```
#include <simplecpp>  
  
main_program{  
    turtleSim();  
    cout << "No. of sides?";  
    int numssides;  
    cin >> numssides;  
    repeat(numssides){  
        forward(10);  
        right(360.0/numssides);  
    }  
}
```

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 nofsides

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

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

Divide the number 360 by the number stored in the space named nofsides 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 **(Sequence)**
 - Retraced if there is a repeat statement **(Iteration)**
 - Later we will see selective execution **(Selection)**
- **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.

The Spirit of The Course

Learn C++ statements/concepts

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



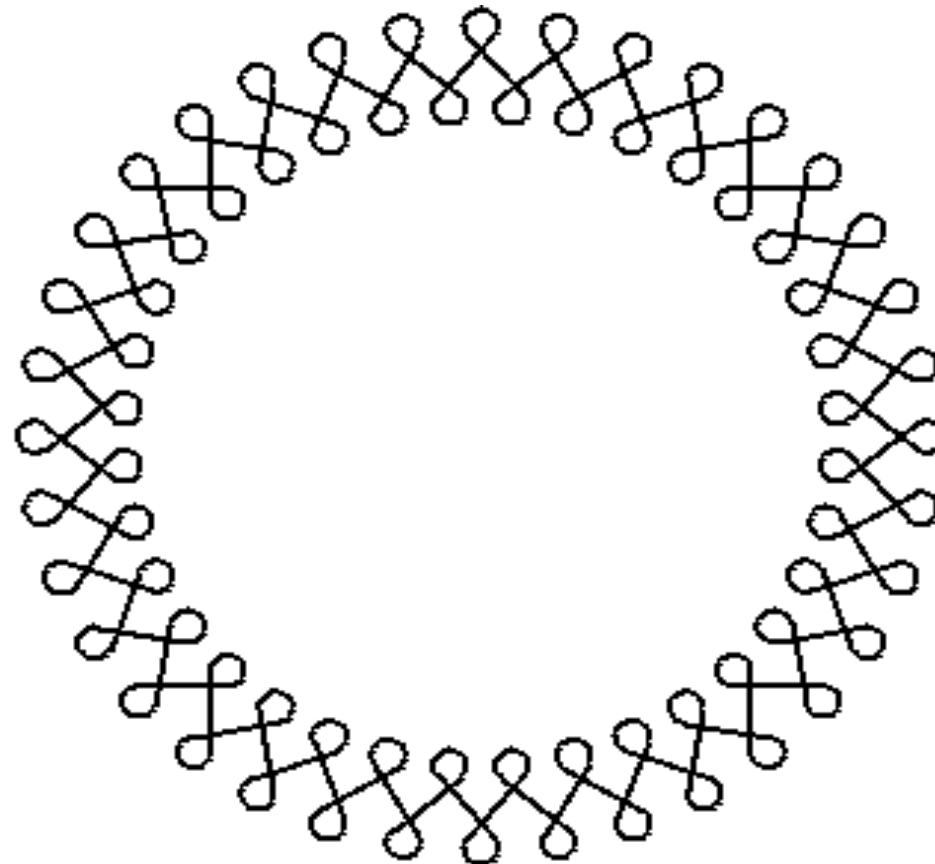
Why Picture Drawing?

- Picture drawing requires calculation
e.g. $360.0/\text{numsides}$
- “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

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



Course Logistics

Course components

- **Course Webpage:**
 - <https://www.cse.iitb.ac.in/~cs101>
 - Has a link to the semester's course plan
- **Lectures:**
 - LA 101,
 - Section S1: Tue, Fri – 3.30-4.55pm
 - Section S2: Wed, Fri – 11.05-12.30pm
- **Labs:**
 - Tue, Wed and Thursdays each week
 - Tue Aug 15th will be held on Fri Aug 18th
 - No Labs during the week of Oct 16th and Nov 6th.
 - SL1, SL2 and Basement 101 Lab of the new CS building.

More on the Labs

- **Ungraded labs (5% of the grade)***
 - Meant for you to practice programming
 - Will have a fixed problem set
 - Will have full TA support during labs
 - No need to submit anything
 - 8-9 ungraded labs and attendance counts towards the grade
- **Graded Labs (40% of the grade)***
 - Lab exams
 - 5 Graded labs for the semester (2 before mid sems and 3 after midsems)
 - Weeks of Aug 7th, 28th Sept 25th, Oct 23rd and Oct 30th.
 - Best 4 of 5 scores will count for 10% each.
 - ***No makeups if you miss any graded labs.***
- ***Lab allocation (student to lab day and lab) is put up on the course home page. Labs start on Tuesday July 25th.***

* *Subject to change at the Instructor's discretion*

Even more on labs - TAs

- Each of you is matched up with a TA in your lab session.
- Your TA is the first source of help for anything in the course.
- The lab allocation also has the name of your TA.
- Find out your TA's email during your first lab.

Fridays Supplemental Instruction

- Help sessions during the lab slot on Friday
 - 8.30-10.30pm at Basement Lab of new CS bldg
 - ***No help sessions on Aug 18th, Aug 25th, Oct 19th***
- Must sign up explicitly for this each week you feel you want to attend it.
 - Signup link is on the course home page.
 - ***Signing up and not showing up will result in a penalty***

SAFE Quizzes

- **In class quizzes** using the SAFE system.
 - Need to download a free app from the Google play store for this.
 - Login with IITB LDAP ID and Password
 - Use Quiz ID to answer the quiz
 - Quiz ID will be given just before the quiz in class.
 - **Worth 5 Percent of the course grade***
 - *Will consider best 75% of the quizzes.*
 - *No Makeups.*
- * *Subject to change at the instructor's discretion*

SAFE Instructions – Its on the webpage

Try out the test quiz live now

1. Download and install SAFE Android app: [**https://
play.google.com/store/apps/details?
id=com.iitb.cse.arkenstone.quizapp**](https://play.google.com/store/apps/details?id=com.iitb.cse.arkenstone.quizapp)
2. Turn OFF cellular data, turn ON WiFi (airplane mode is best)
3. Open SAFE app, read instructions
4. Login using credentials given by instructor
5. Wait at Quiz-ID screen, enter it when instructor announces it
6. Attempt quiz: use previous/next buttons, navigation
7. Submit when done attempting all questions you can
8. On timer expiry, app will auto-submit
9. Wait until successful submission
10. Login again, press ‘previous submissions’ to see marks/
answers

Written Quizzes and Exams

- **2 Quizzes, each 1 hour long.**

- Fri 18th Aug, 8.15am to 9.15am
 - Fri 13th Oct, 8.15am to 9.15am
 - Timings/dates may change.
 - ***Worth 5% of the grade EACH****
 - ***No makeups.***

- Midterm and Final Exam

- ***Dates/timing TBD***
 - ***Midterm worth 15% of course grade****
 - ***Final worth 25% of course grade****

*** Subject to change at the instructor's discretion**

Prutor

- Programming IDE
- Supports both turtlesim for drawings as well as regular C++ programs.
- Is the ONLY interface to all lab assignments.
- TAs will do a tutorial in the first lab.

Piazza – Your friendly neighborhood help desk

- Ask ANY question and ye shall be answered
 - Within 24 hours, probably a lot sooner.
- You are already enrolled and must have received an email about it.
 - If you have not, see your Lab TA about it during the first lab. OR
 - https://piazza.com/iit_bombay/fall2017/cs101
 - Access code is cs101
- Bookmark this link:
https://piazza.com/iit_bombay/fall2017/cs101/home
 - And use it for anything related to the course.
 - There are folders for each aspect of the course – graded labs, ungraded labs, quizzes, lectures etc.

A Note on Cheating and Academic Dishonesty

- Absolutely no form of cheating will be tolerated
 - Unless otherwise instructed (to work in a team) you MUST turn in only work that you have done on your own for any evaluation in the course.
- **Cheating =>**
 - **Fail grade (no exceptions)**
 - **A report to Dean of Students**