

# CS 101 Computer Programming and Utilization

## Lecture 1 Background and Introduction

Tue Jan 4, 2010 2:00-3:30, FCK Auditorium  
Wed Jan 5, 2010 11:00-12:30 PCS Auditorium

*Prof. R K Joshi*  
*Computer Science and Engineering*  
*IIT Bombay*  
Email: [rkj@cse.iitb.ac.in](mailto:rkj@cse.iitb.ac.in)

# What is Programming?

- Training a machine to solve our problems
  - Look at this usage of the term 'programming':
    - “the brain is programmed”
  - By this we mean that the brain is “trained” to do certain things, or to behave in certain predictable ways. Here the brain is the machine.
- A program represents knowledge about doing something, and a machine stores this knowledge and applies it to a given situation.

# Examples of programming in commonly used devices ..

- Fan Regulators
- Accelerators
- Applying brakes
- Fail safe train signaling
- Periodic road traffic control signals
- Pressure cooker safety valves
- Remote Controls
- Vending Machines
- Telephones

# Characterizing programs

- Input
- Output
- Logic that produces output from input
  
- Identify input, output and the logic in the examples given on the previous slide?

# Complex Programs

- A program can have many inputs and many possible outputs
- For example, object 'telephone' has a complex logic
  - Keypad, connection and conversation, dial tone, busy tone, ringing tone, timeouts, call transfer, caller identification...
  - Programs in cell phone devices have applications too.
- Similarly, speed of a vehicle depends not only on accelerator position, but also on gear position, load in the vehicle etc. These additional parameters are part of the system's state..they are *hidden inputs* in some sense..

# Handling Error Conditions

- Programs need to work in fail-safe mode.
  - Example of failure of automatic door locking system in cars
  - A number conversion error caused a rocket failure due to overflow.. and the redundant unit also failed.. guess the reason?
- Many more output possibilities need to be considered

# Some examples of computer programs that you may have used

- Withdrawals through ATM
- Online Train Reservations
- Online Banking, money transfer
- Desktop cricket scores
- Searching through the Internet
- Computer Games
- Media players
- Social networking sites

# Some example applications from Engineering and Scientific Computing

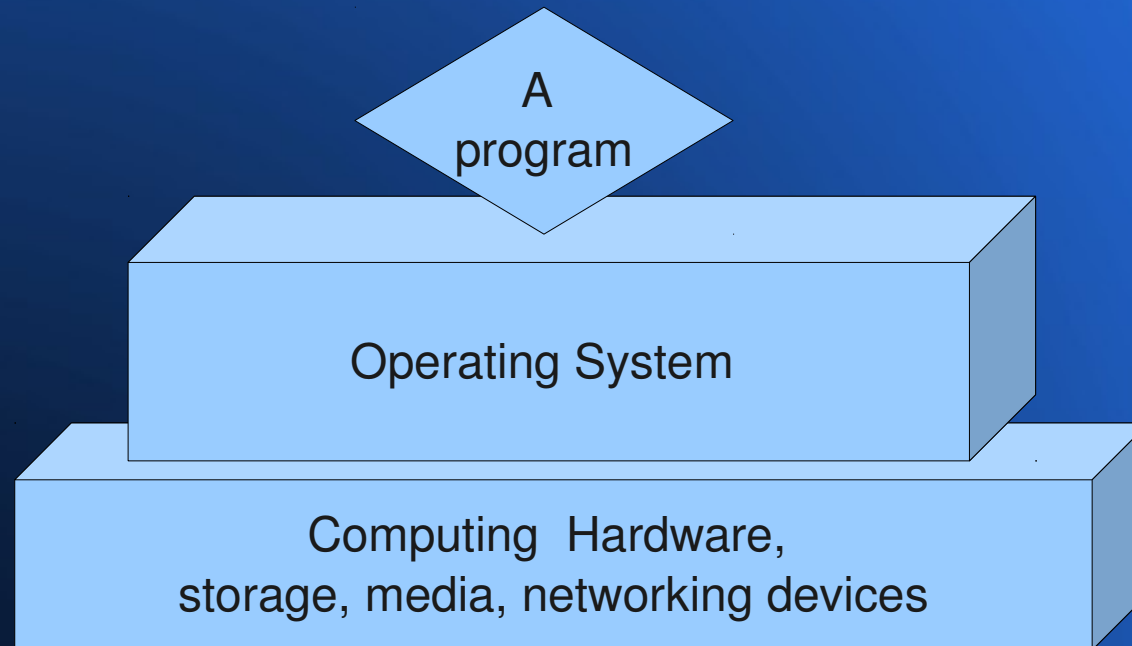
- Design of machines and simulators
- Weather forecasting
- Image recognition
- Bioinformatics
- Speech and sound analysis
- Natural language processing
- Mathematical tools and programs
- Design and processing software



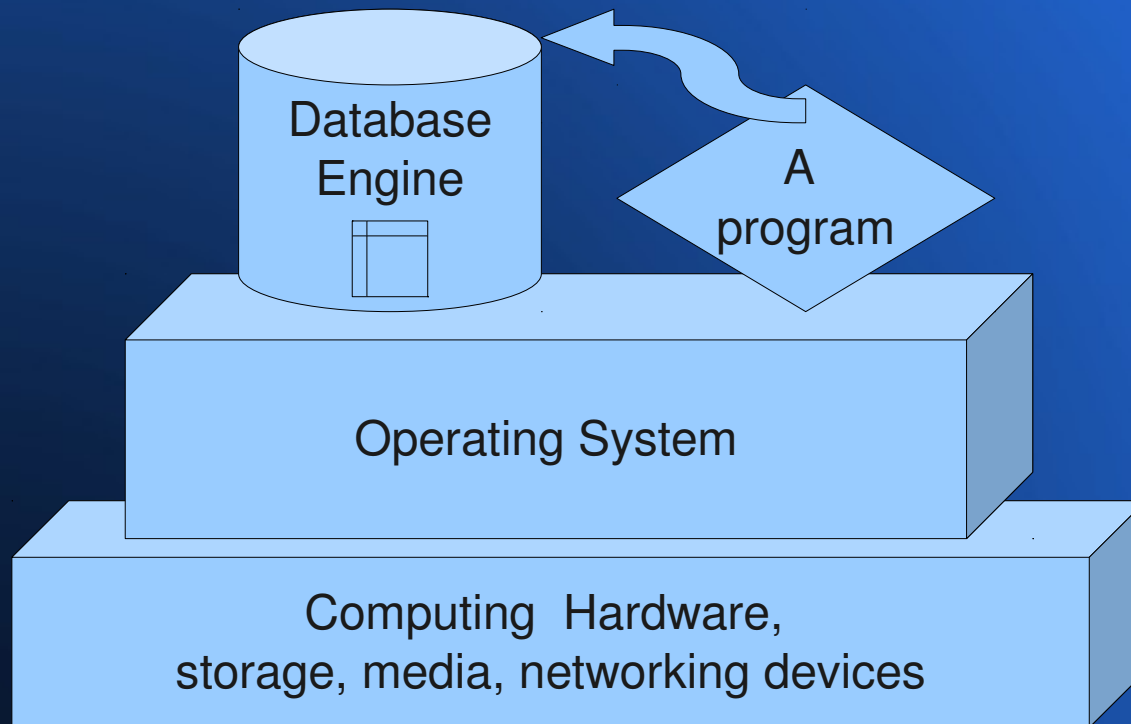
# Common Programs for programming and utilization of computing systems

- Programming language compilers and interpreters
- Operating systems
- Device drivers
- Database Engines
- Table viewers and manipulators
- Document editors
- Presentation editors
- Commands that work on command line
- Html editors
- Integrated Development Environments

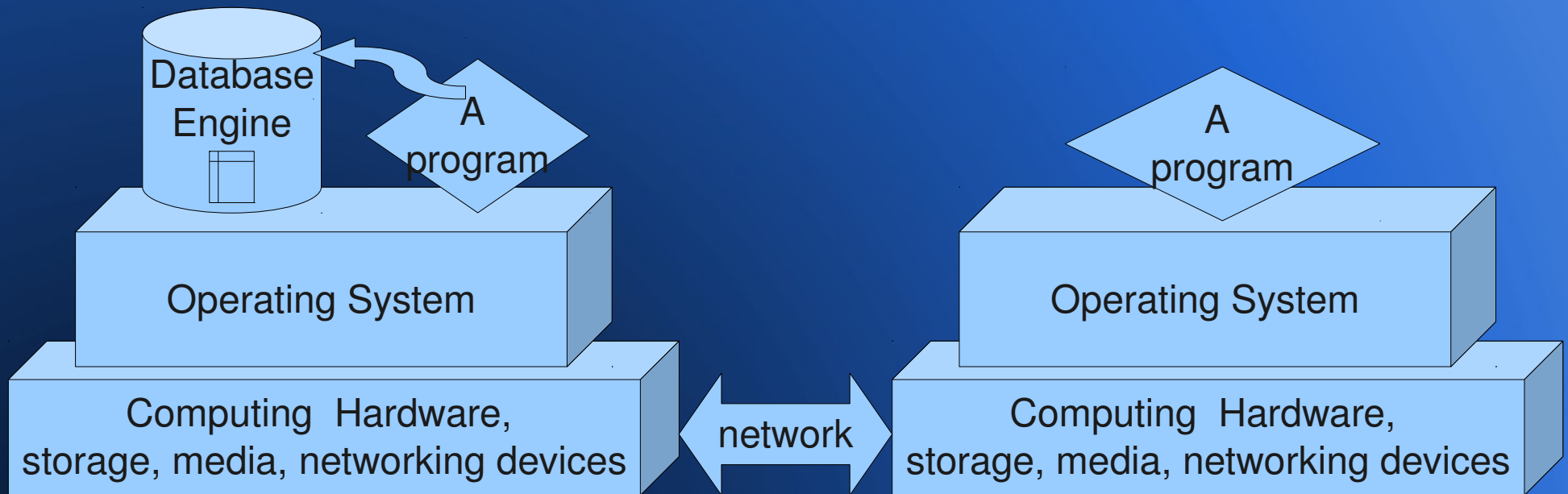
# A typical computing application



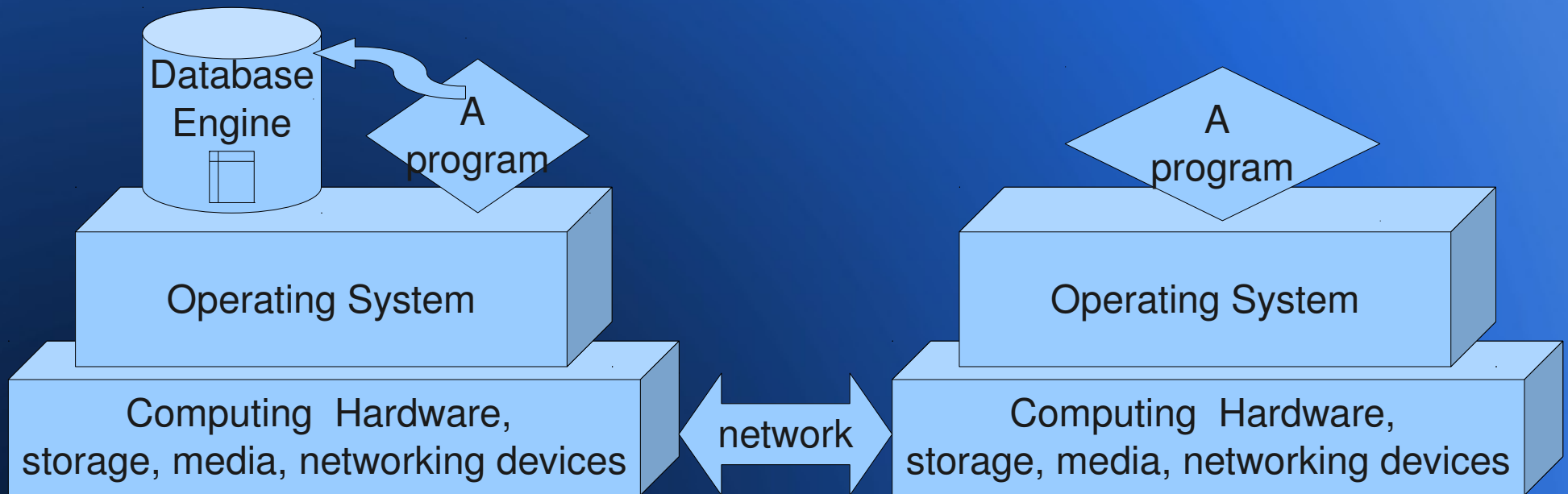
# A more advanced application ..



# A more advanced application ..



# Everything in this picture is associated with programming..



Haven't you used such a system earlier?

# Why Do we study Computer Programming and utilization?

- Introduction to
  - Using computers
  - Developing programs or computations
- Basics of computer programming
  - You can learn advanced programming once you have understood basic programming principles
- Applications of programming are all around:  
engineering, sciences, management, business, entertainment ..

# Programming needs discipline

- In absence of it:
  - Programs may not be fool-proof.
  - Programmers spend more time in debugging
  - Programs developed by one person are not easily understandable by another
  - Programs developed by one person are not easily usable in other programs of even the same person
  - The developer may get confused by one's own programs!
- Are these problems solvable by appointing more such programmers?!

# The reusability aspect

- Can you use the heat generated inside a car bonnet to toast a bread when the car is running?
- A car has dynamo and battery too.. so can that power be used to play music?
- Can the same power be used to operate a vacuum cleaner?
- Reuse needs good packaging and some standardization
  - e.g. Nuts and bolts
- Programs also need to be 'packaged' nicely.. programming languages provide the required constructs.



# Steps in Program Development

- Programming is all about the logic that goes into a program
  - You need to think about it first, and note it down
- Then translate it into a program written in a programming language like C++
  - Type out the source program on a computer.
  - Save it
- That program is then 'compiled'
  - Check the 'syntax'
  - Produce an 'executable' version
- Execute it (Run it!) and have fun..
- Look out for bugs.. test the program and debug it to fix the bugs you find.

# Some Characteristics of Good Programs

- Well dressed -- well indented, good looking, use of spaces, consistent notations
- Well informed – use of comments
- Correct – well tested or proved to be correct
- Clarity – the logic is expressed clearly in terms of well selected and well structured programming primitives

# One more example ..

- A program that finds out whether a given number is prime
- Think about input, output and logic?
- A programming language has constructs to express all these ..

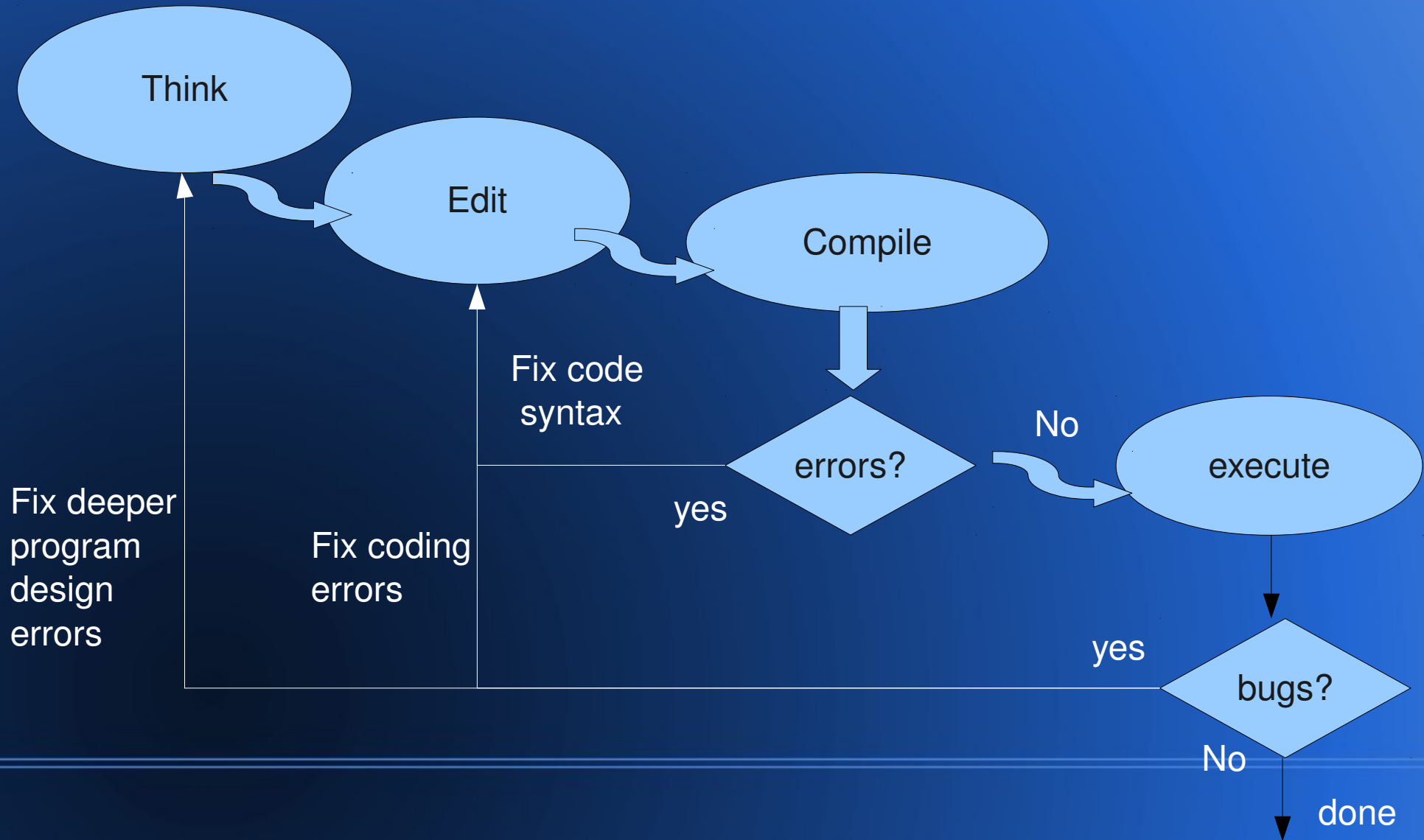
# A Programming Language

- Like a natural language, it has dictionary words, you can create new words for indicating values, there are statements and there is grammar too.
- Provides primitives to express the logic of programs
  - Data or values
  - Control or logical steps
- Numbers, characters, strings..
- If then decisions...
- And more ..

# About the course

- Utilizing computers
- Programming in C++
  - Various programming constructs
- Applying Programming
- Programming Assignments, Labs

# Let's try to understand this cycle of program development..



# Observe the activity boxes and the forward and backward arrows..

We will come back to it when we will actually develop programs.

Many times you will end up combining the first two activities while sitting in front of the computer.. So you may miss the distinction between the two leftmost upward arrows..watch out!