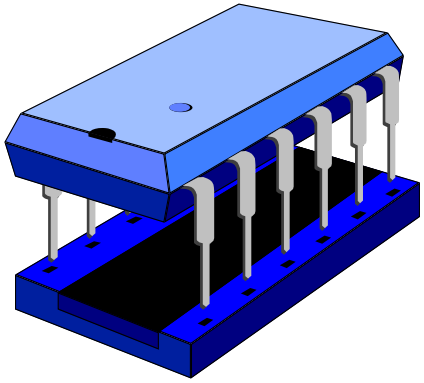
COMPUTER ARCHITECTURE



Introduction

Bernhard Kainz (with thanks to **A. Gopalan, N. Dulay** and **E. Edwards**)

b.kainz@imperial.ac.uk

Course Aims

- To understand the elements and functional principles of a computer

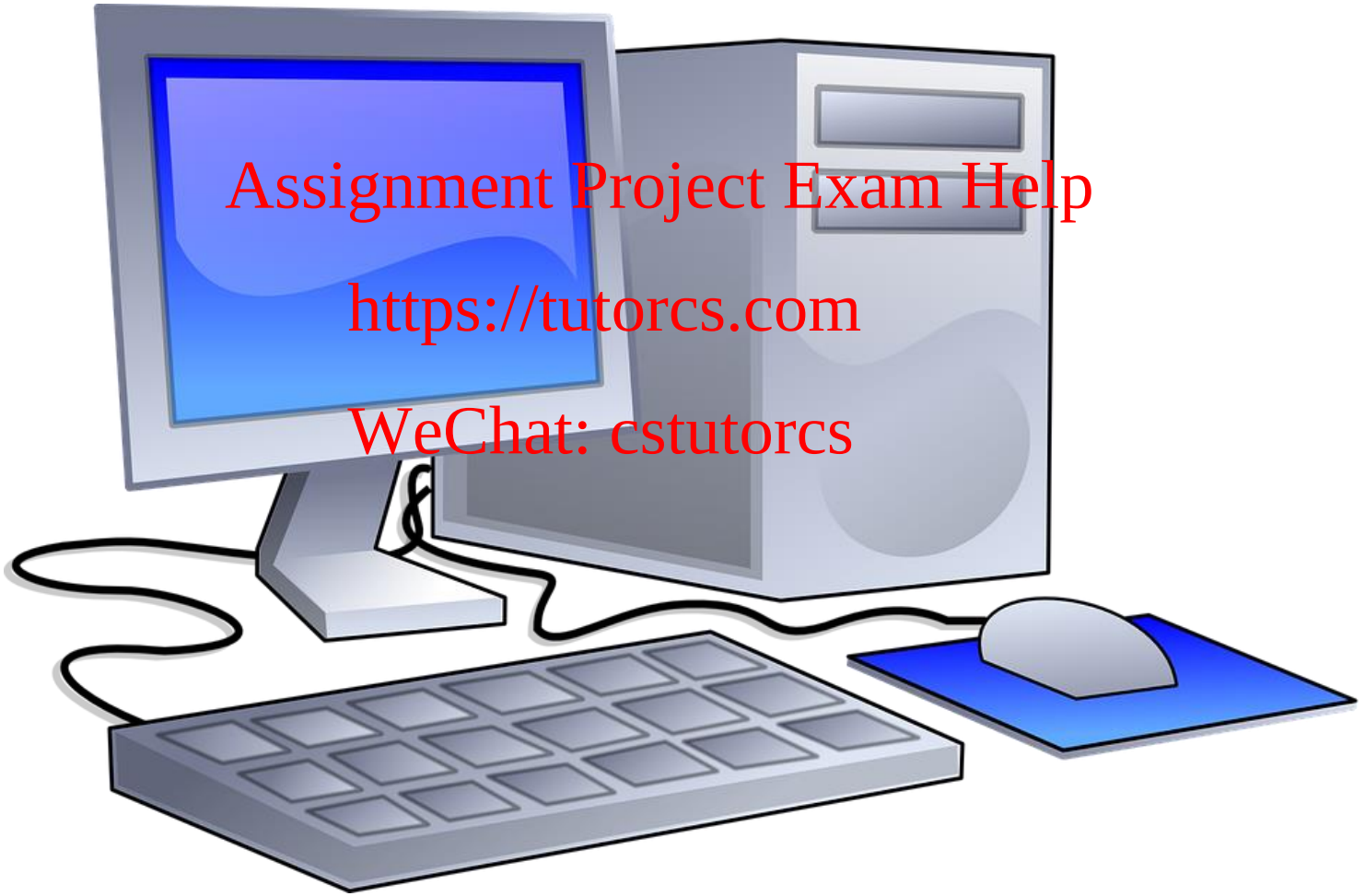
Learning Outcomes

- At the end of this course you should:
  - Know the basic the elements of a computer and understand how these elements link together
  - Know the different forms of memory organisation
  - Understand the basics of logic and number representation
  - Comprehend the different levels of programs
  - Understand the structure of the Toy and Pentium processor
  - Be able to write assembler programs

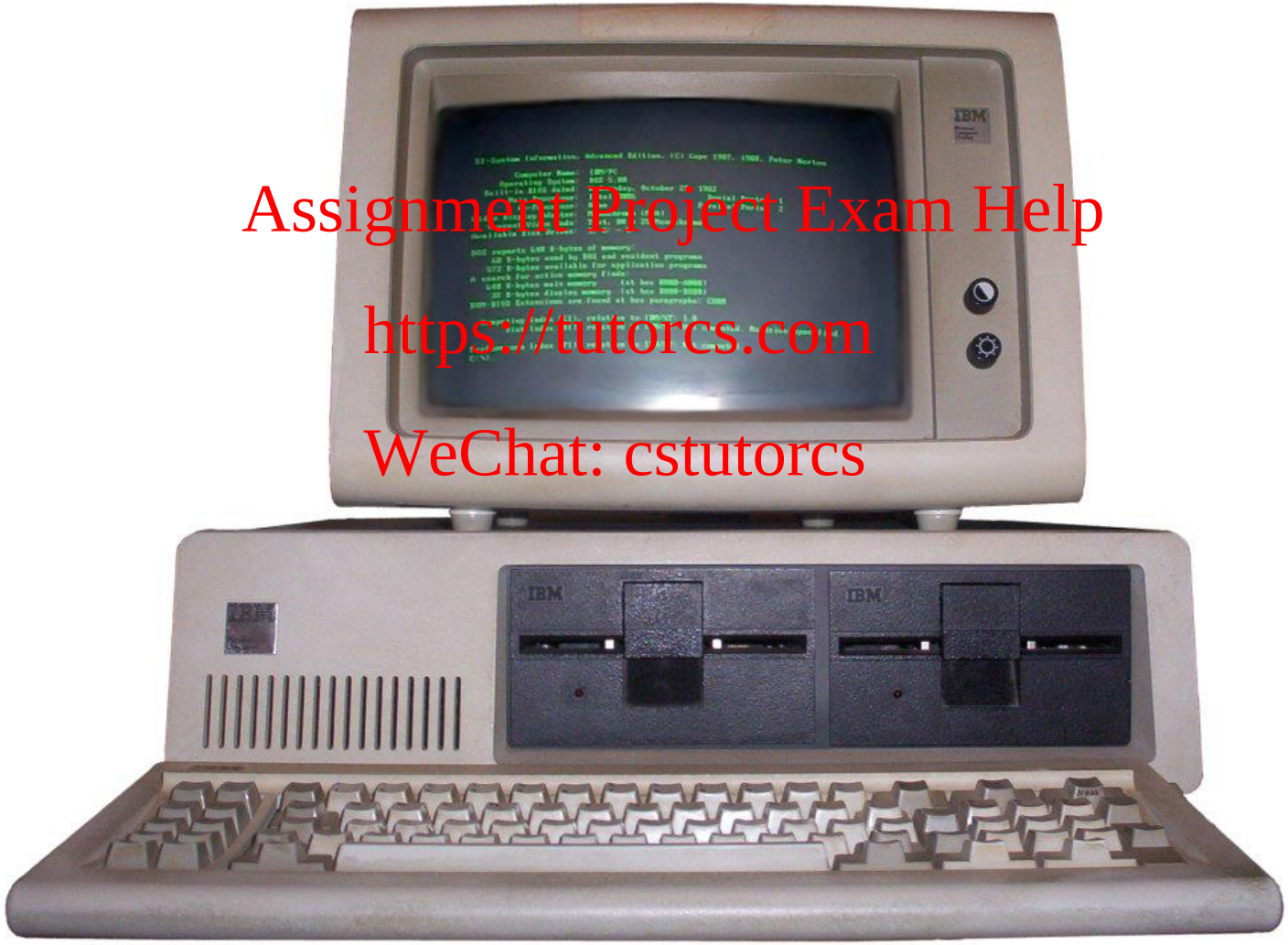
What is a computer?

Pixabay -- free for non-commercial use

What is a computer?



What is a computer?

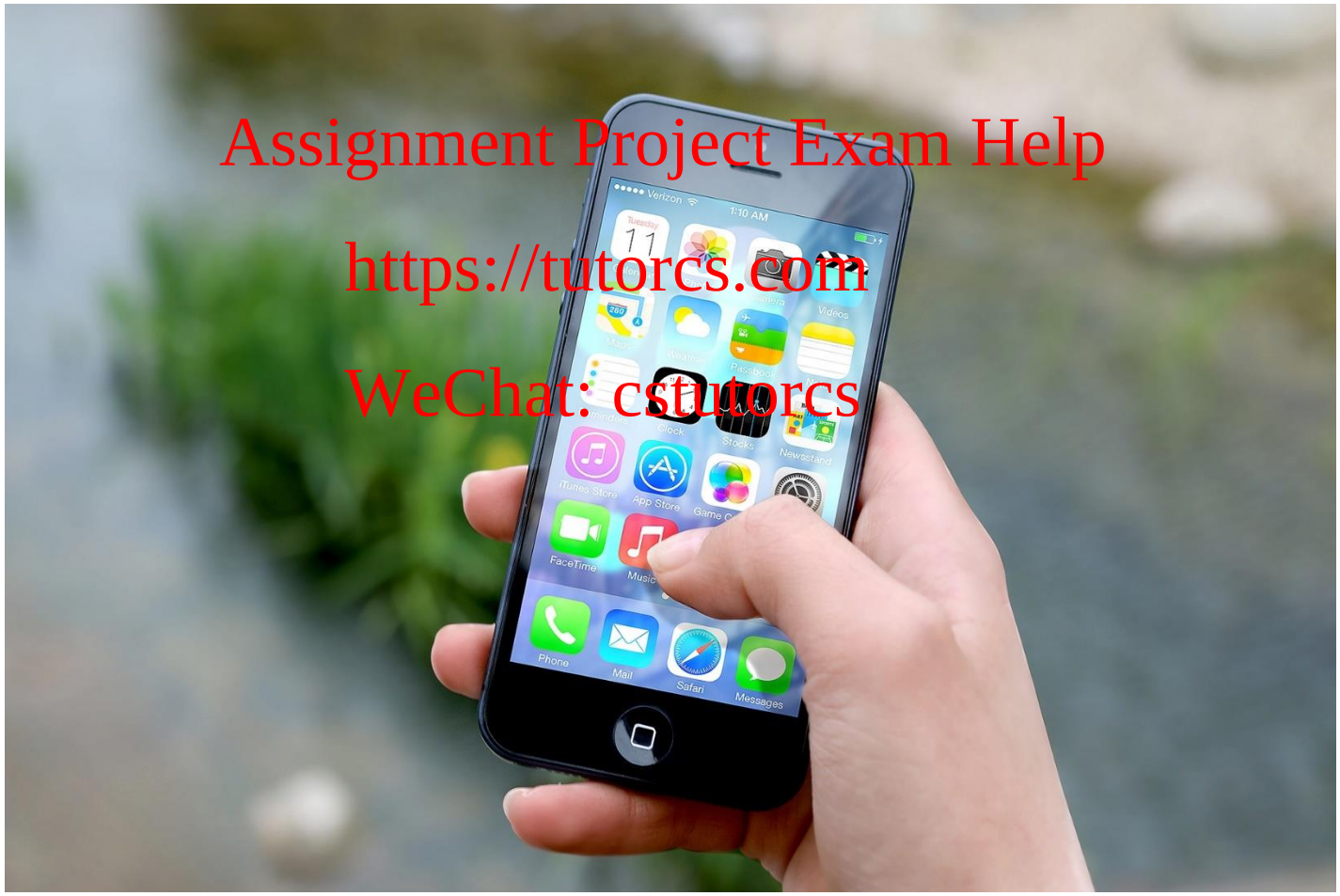


wikipedia -- free for non-commercial use

What is a computer?

wikipedia -- free for non-commercial use

What is a computer?



flickr -- free for non-commercial use

What is a computer?

What is a computer?

What is a computer?

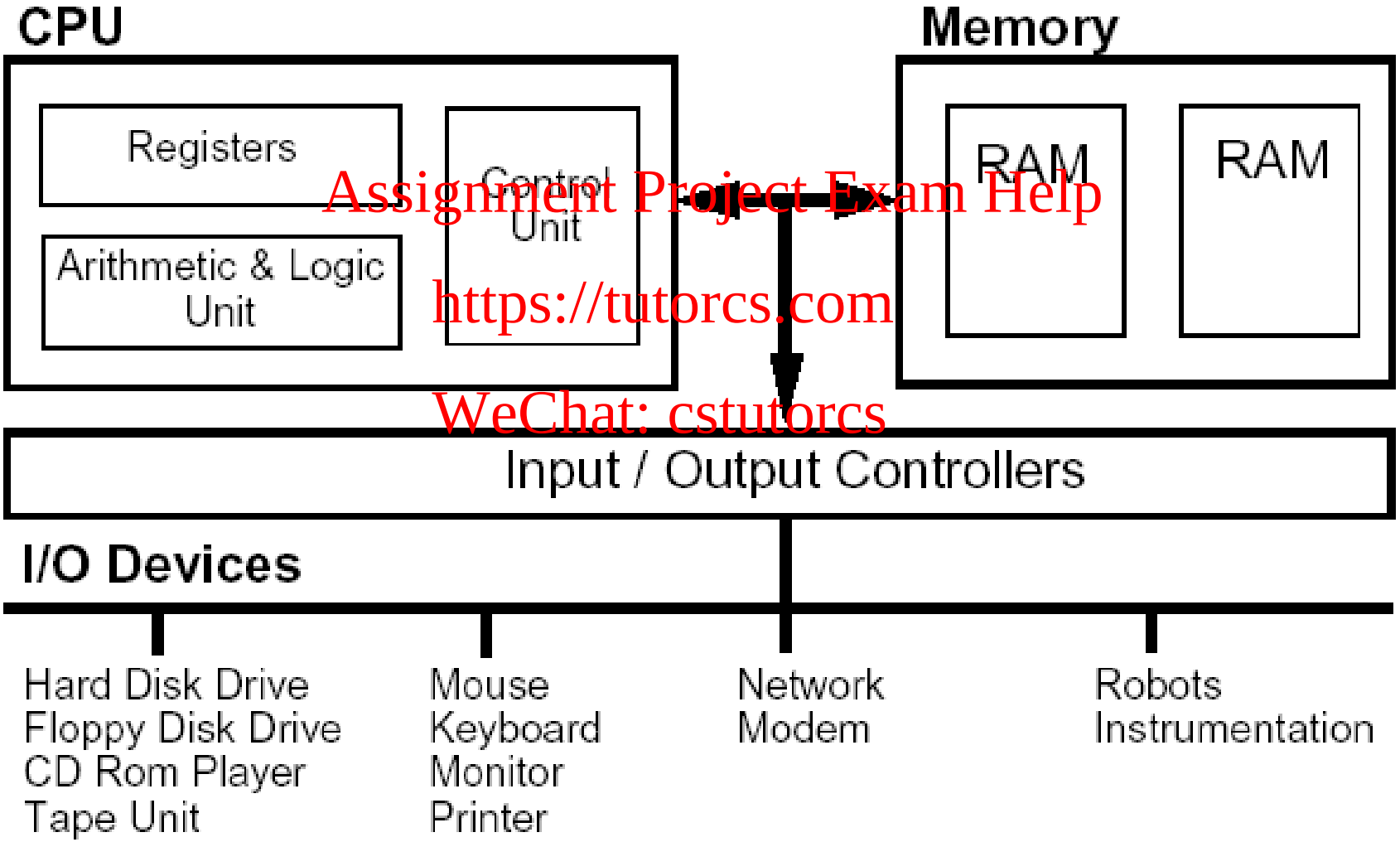
flickr -- free for non-commercial use

What is a computer?

What is a computer?

flickr -- free for non-commercial use

What is a computer?

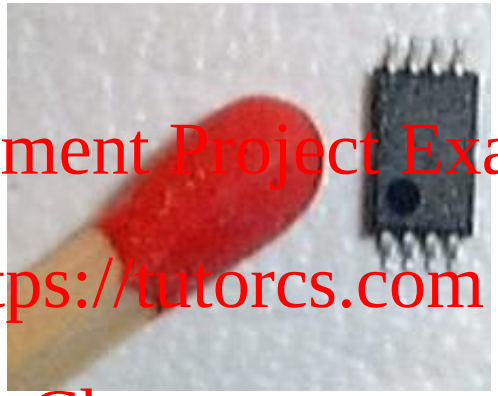


What is a computer?

Nvidia -- free for non-commercial use

Downsizing

Matchbox computer

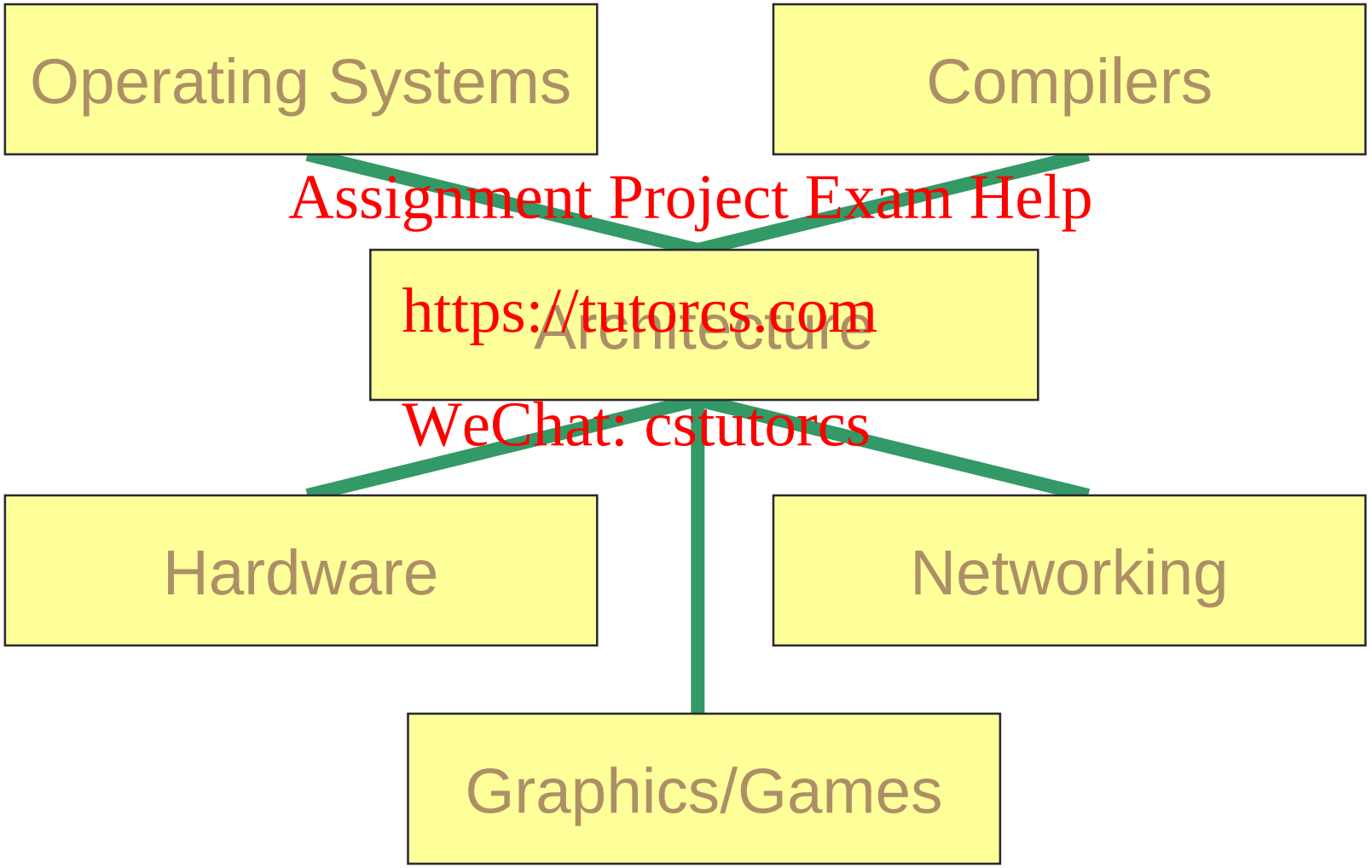


Web Server

UC Berkeley Mote


Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Why ?



Computer Architects - What do they do?

- **Instruction Set** Design
- **CPU** Design
- **I/O Interface** Design
- **Bus design**
- **Motherboard** design
- ....

- **Emulation & testing** of the Architecture in Software
- **Implementation and testing** of the Architecture in Silicon
- **Performance** Evaluation

- Requirements with input from:

  Higher Management,
  Compiler writers,
  Operating System developers,
  Sales and marketing,
  Existing and potential Customers

- **Cost/profitability** analysis

Computer Architecture - Who needs to know?

- **Students** of Computer Architecture!
- **Lecturers** of Computer Architecture !!
- Operating System Developers
- Compiler Writers
- **Repair and Maintenance Technicians**
- **Third Party Vendors** e.g. Peripheral makers, Memory suppliers, Add-on card Suppliers (e.g. Co-processors, Graphics Accelerators)
- Sales and Marketing
- **Patent Office Workers**
- Reverse Engineers/Hackers

Course Outline

Part 1

Boolean Algebra and Logic

Basic Circuits and Memory

CPU Design

Main Memory Organisation

Data Representation & Binary Arithmetic

Floating Point Representation

Part 2

CPU Organisation & Representation

Pentium CPU and Programming

Input/Output Control

Recommended Reading

Structured Computer Organisation (5th ed.)

- Andrew S. Tannenbaum, Prentice-Hall International
- Easy to read, also covers 2nd & 3rd year topics

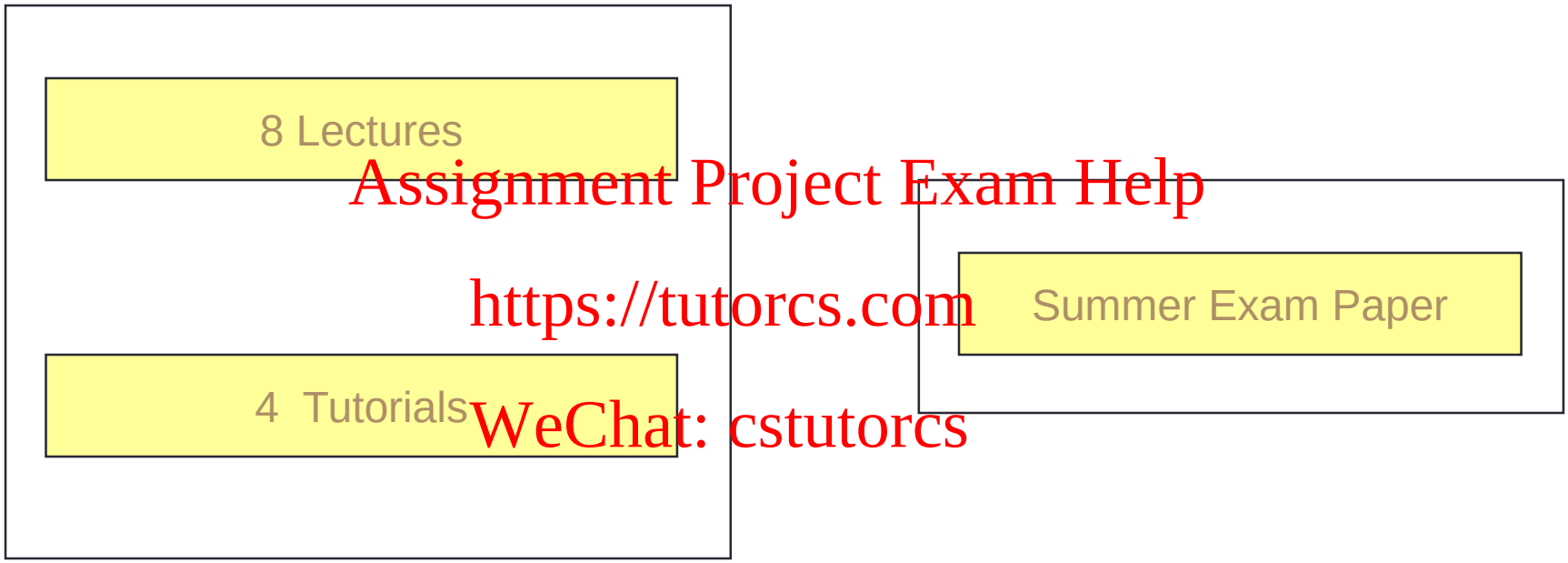
Computer Organisation & Architecture (9th ed.)

- William Stallings, Prentice-Hall International
- Detailed, academic, also covers 2nd and 3rd year topics

Guide to Assembly Language Programming in Linux

- Sivarama Dandamudi, Springer
- Good introduction to Intel assembly programming

Workload (Architecture – Part 1)



- Lecture notes and Tutorials available from CATe
- Questions and discussions on Piazza
- Coursework on paper, 2 weeks time

Schedule

Date	Topic	Lecture/Tutorial	Time and Room
October 20^{th}	Introduction + Boolean Algebra and Logic	Lecture	11:00 - 12:00, 145
	Boolean Algebra and Logic	Tutorial	12:00 - 13:00, 145
October 23^{rd}	Basic Circuits and Memory	Lecture	14:00 - 15:00, 311
	Chip Design	Lecture	15:00 - 16:00, 311
October 25^{th}	Memory Organisation	Lecture	11:00 - 12:00, 145
	Memory Organisation	Tutorial	12:00 - 13:00, 145
October 27^{th}	Data Representation + Binary Arithmetic	Lecture	11:00 - 12:00, 145
	Data Representation + Binary Arithmetic	Tutorial	12:00 - 13:00, 145
October 30^{th}	Floating Point Numbers	Lecture	14:00 - 15:00, 311
	Floating Point Numbers	Lecture	15:00 - 16:00, 311
October 31^{st}	Floating Point Numbers	Tutorial	14:00 - 15:00, 144
	Coursework released - Deadline November 16^{th}		
November 1^{st}	Tricks and Revision	Lecture	11:00 - 12:00, 145