Welcome to COMS10002

- Lecturer/Unit Director
  - Oliver Ray

- Lecturer
  - Mike Fraser, Ian Holyer

- Demonstrators
  - Rob Frampton and team

- All information is (or will be) on-line:
- `http://www.cs.bris.ac.uk/Teaching/Resources/COMS10002/`

Lectures etc

- Lectures
  - Mon 15:00 (Chemistry, LT2), Wed 10:00 (Chemistry, LT2)
  - Explain theory, live programming, hints on lab work.
- Laboratory sessions (MVB 2.11, Tuesday 9-12am or 2-5pm, Thursday 4-5pm or 5-6pm)
  - Thursday lab is for Worksheets, Tuesday lab is for feedback and courseworks
  - 1 exercise per week
  - Viva at the end, no theory exam!

Preferred contact mechanisms

- Inside lab times (feedback, coursework): Talk to lab assistants

- Outside lab times, please use CS forum (Varun and each other)
- https://www.cs.bris.ac.uk/forum/index.jsp?title=COMS10002

- Chat with Ian Holyer (C).
- https://www.google.com/calendar/embed?src=csijh@bristol.ac.uk

- Chat with Oliver Ray (Haskell).

- Mike is very hard to contact outside lectures (by email, in person, or using smoke signals)

Feedback

- With 160 of you and only a handful of us, feedback is important
- Most ’courseworks’ are formative/unmarked. A few are marked (see SAFE)
- Verbal feedback in labs for worksheets, courseworks and exams
- If you do not understand, please use the labs to get feedback
- We provide opportunities for feedback but the responsibility to take them is yours. If you do not ask, we will assume you understand

Finally ...

Please tell us

- of any improvements that we can made
- which parts need more attention, or less attention
- if you are lost or bored
- If you want to take the first exam early and skip the early courseworks, we can provide more challenging problems to keep you entertained. Talk to Rob in your first Tuesday lab. Be warned, this is not for a faint hearted.

What is COMS10002 about?

Programming and Algorithms

- What do we use a computer for?
- What does a computer consist of?
- What does it do?
- How to tell it what to do?

Possible objectives:

- You will be able to use some standard computer packages
- You will be able to program computers
- You will know how the computer works
- You will know some optimal algorithms and data structures for common problems

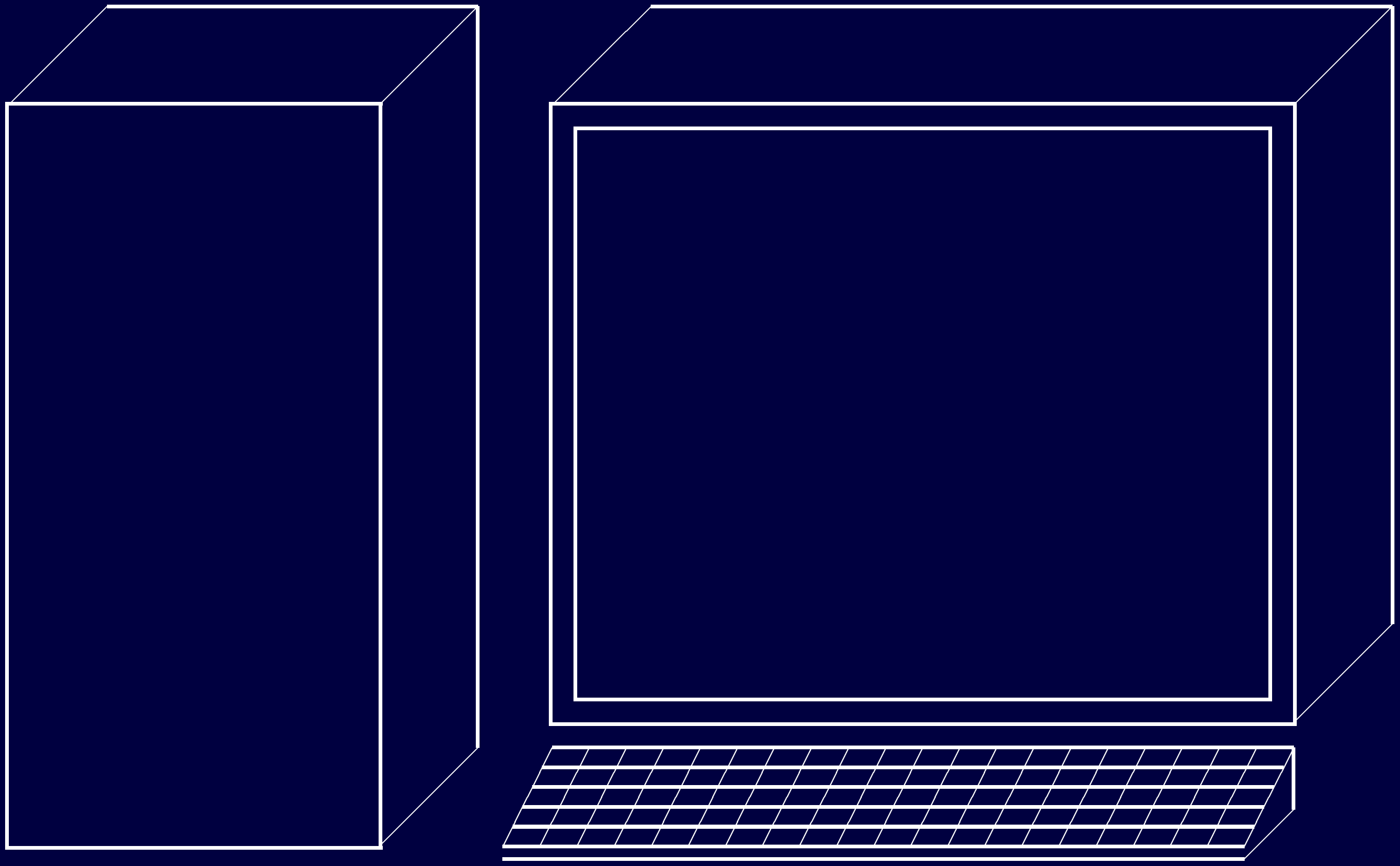
Each of you will have your own objectives

What do we use computers for

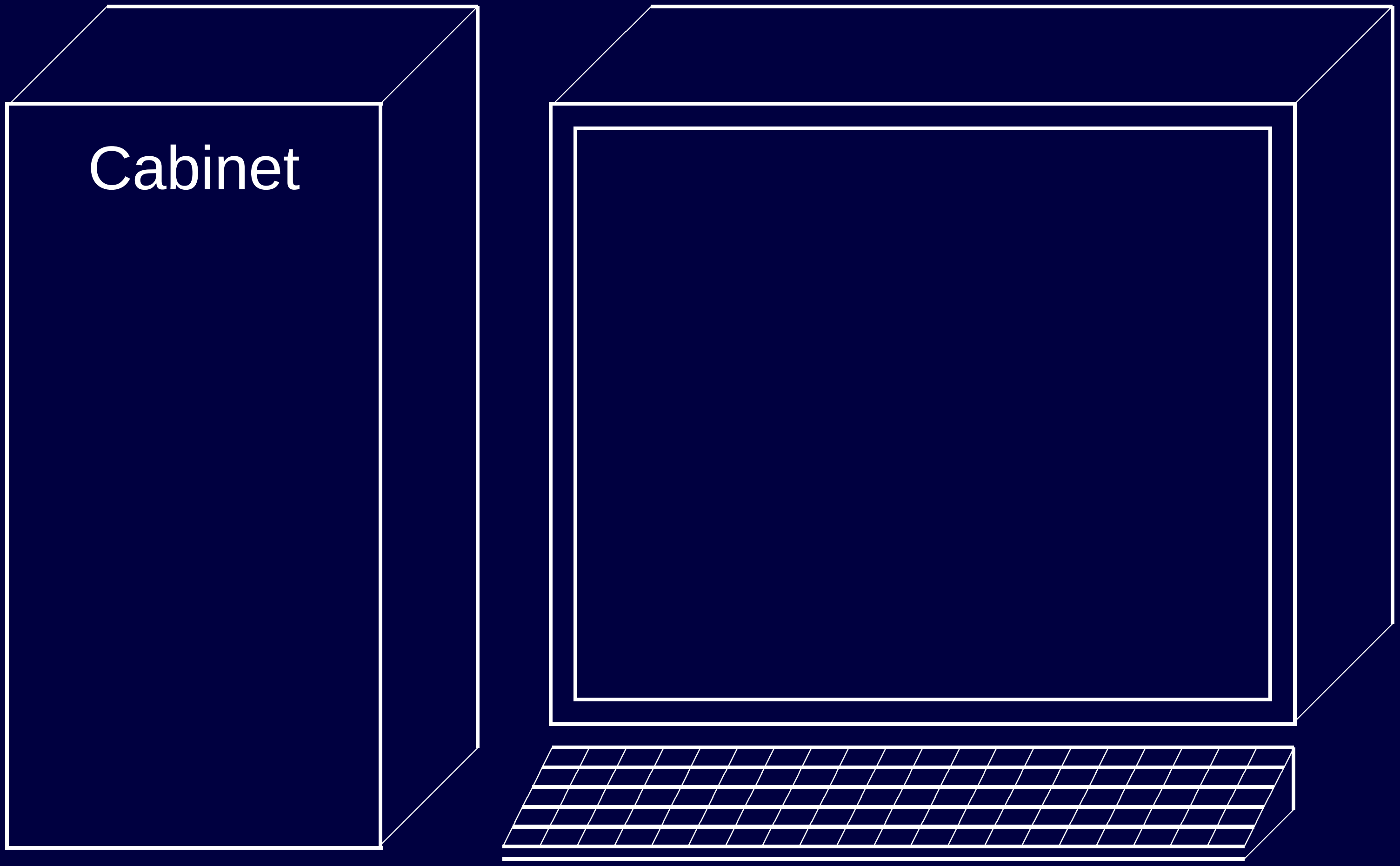
Almost everything.

- Traffic lights, cars (tens: ABS, ignition, stereo, ...), printers, photo-copiers, television, stereo, microwave oven, mobile phone, ATM, video, X-ray machines, cash-registers, watches, design (CAD / CAM), maths, physics, ...
- Even though some should work perfectly well without one.

So, what is a computer (desktop)?

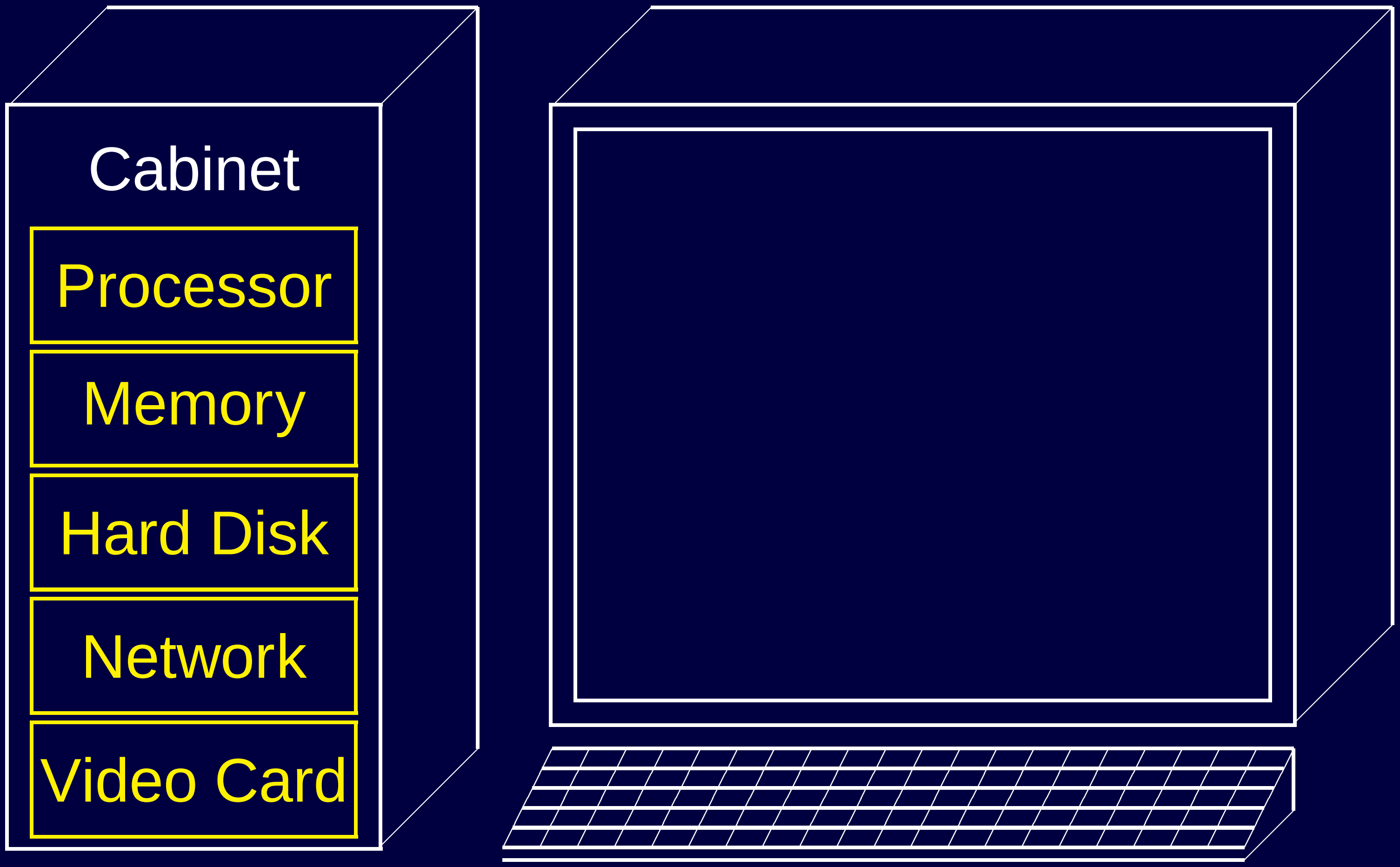


So, what is a computer (desktop)?



A Cabinet, Screen and Keyboard

So, what is a computer (desktop)?



A Cabinet, Screen and Keyboard

A CPU, Memory, Permanent Storage, Networking, Graphics

- Known as the *Hardware*

Summary of the Hardware

Processor (CPU, Central Processing Unit):

- The heart of the machine. “Executes” any program.

Main Memory:

- The memory of the processor, it stores all volatile data.

Permanent storage (Hard Disk, CD-Rom):

- Stores data for longer time: your address book.

Network (TCP/IP):

- Interfaces computer to the other computers.

Devices for Human Computer Interaction

- Keyboard, Screen, Mouse
- Speaker, Camera, Microphone, Scent-sensor
- Not in *embedded systems*, such as a car or a toaster.

The hardware executes software

Software consists of the programs running on your computer.

- The difference between an ABS and a mobile phone is the software.

ABS, mobile phones etc typically execute just one application

General purpose computers can be told to run any application:

- Simple computers (MS-DOS):
  - Execute one program at a time
- More complex computers (earlier Windows, earlier MacOS):
  - Execute multiple programs at a time, but only one user.
  - Multi programming, single user.
- Real computers (Windows, Unix, VMS)
  - Execute multiple programs at a time, for multiple users.
  - Multi programming, Multi user.

MS-DOS, Windows, Unix are *Operating systems*

Operating Systems

So what is an operating system?

- An operating system is a program which runs on every computer
- It is the mother-of-all-programs.
  - ⇒ Operating system is listening to keyboard and mouse
  - ⇒ Operating system is starting and stopping other programs
  - ⇒ Operating system maintains permanent storage

⇒ The operating system *manages* your computer hardware; it is in control

Core of the operating system?

- The idle loop!

Simple operating system

```
begin:
   Has the user hit the return key?
   If not go back to begin.
   Assume user typed the name of a program
   Locate the program
   Load it into memory
   Execute this program

   Remove the program from the memory
   Go back to begin.
```

- The actions of an operating system

Simple operating system

```
begin:
    Has the user hit the return key?
    If not go back to begin.
    Assume user typed the name of a program
    Locate the program
    Load it into memory
    Execute this program

    Remove the program from the memory
    Go back to begin.
```

- The IDLE loop (computer does mostly nothing).

Simple operating system

```
begin:
    Has the user hit the return key?
    If not go back to begin.
    Assume user typed the name of a program
    Locate the program
    Load it into memory
    Execute this program

    Remove the program from the memory
    Go back to begin.
```

- Locate and load the program

Simple operating system

```
begin:
    Has the user hit the return key?
    If not go back to begin.
    Assume user typed the name of a program
    Locate the program
    Load it into memory
    Execute this program
     ... This may take a while ...
    Remove the program from the memory
    Go back to begin.
```

- “Add these two numbers”
- “Tell me if this bridge will collapse”
- “Fly this airplane to Cuba”

Simple operating system

```
begin:
    Has the user hit the return key?
    If not go back to begin.
    Assume user typed the name of a program
    Locate the program
    Load it into memory
    Execute this program

    Remove the program from the memory
    Go back to begin.
```

- Free the memory

Which questions will this unit answer?

How can I write a program?

- Basic Software Engineering
- Coding (a language to program in)
  - You will learn this by *doing it*
  - That is the only way to learn programming and problem solving.
  - Studying books only helps after you have cracked it.
  - You have to take the first hurdle yourself.

How is the program executed?

- Hardware (Explore the processor and memory)
- Software (Operating system, and other standard software)
  - You will learn this in lectures
  - Can read up on it in a textbook.

Assessment

This is a 20 credit unit

⇒ On average you should spend 16 hours a week on this unit.

- 2 hours lecture, 4 hours scheduled lab, 10 hours preparation/unscheduled lab.

Computer Science is a practical subject

⇒ Most of the marks (and learning!) come from lab work on this unit.

- Group work is not permitted for summative assessments, of course it is allowed to discuss work with your friends.
- Plagiarism is a waste of your time and money. If you can't do it, please ask us otherwise you will get stuck and fail.

Summary

Programming begins with practice, only then does the theory make sense.

Software executes on hardware, it is important to understand the relationship.

There are many possible ways to solve problems, some good rules of thumb and optimal solutions are available for common ones.