LOYOLA UNIVERSITY CHICAGO

CENTER FOR TEXTUAL STUDIES AND DIGITAL HUMANITIES

DIGH 401 - Introduction to Computing

Fall Semester 2014

Week 1

Course Schedule

- No class on Tuesday 7th October 2014

- Last class on Tuesday 2nd December 2014

- Final assessment due on Thursday 11th December 2014

Course Details

- Instructor: Dr Nick Hayward

- Email: nhayward@luc.edu or ancientlives@gmail.com

- Office: Loyola Hall 316

- Office hours: Tuesday 8am-9am and 2pm-3pm

Goals of the course

- introduce computer systems, their development, and the impact of computers on society

- application of computing and computer science in academic research, publishing, libraries, and the arts

- consider history of programming and its implementation

- explore theory of programming and its practical application

- procedures, analysis, and problem solving

Assessment

Ongoing

- assessment & practical work (30%)

- class presentations (30%)

Final

- conceptual design specification (40%)

Assessment

ongoing assessment & practical work (30%)

- exercises to test material discussed in class

- basic design and coding work

- practical tests of material discussed in class

- assessed primarily on success of programmatic solutions, but also structure, comments, documentation...

Assessment

class presentations (30%)

- scheduled for various intervals during the course
- short 10 minute presentations on pre-defined material
- technical presentation/demonstration & pitch to potential client
- coordinate with each other to avoid repetition

Assessment

Conceptual Design Specification (40%)

- software design specification
- complementary to conceptual project design in DIGH 400

Course website and resources

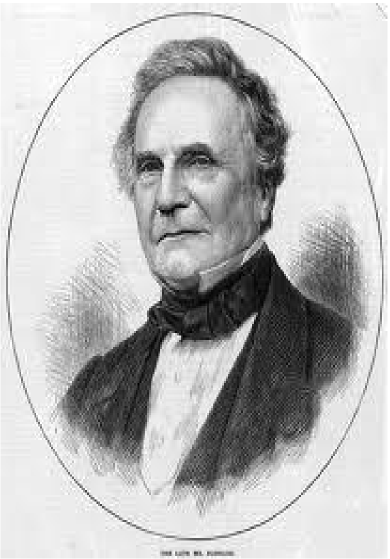
http://students.ctsdh.luc.edu/classes/401/

- course information, timetable and guidelines
- weekly updates
- assignments and weekly work
- latest news
- resources and bibliography

NB: again, it's a work in progress!

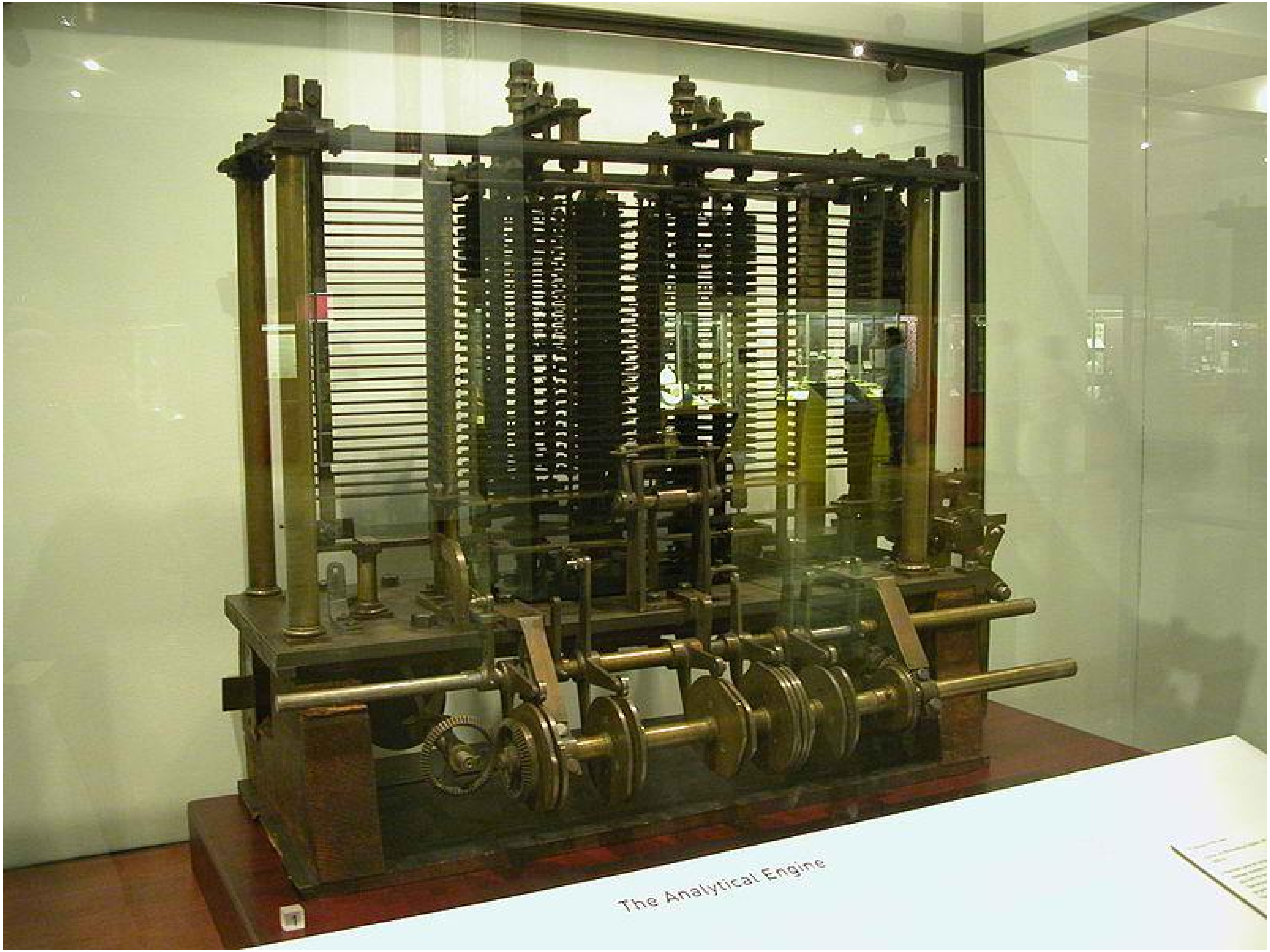
Any questions?

First computer and programming



1834 - Charles Babbage's 'Analytical Engine'

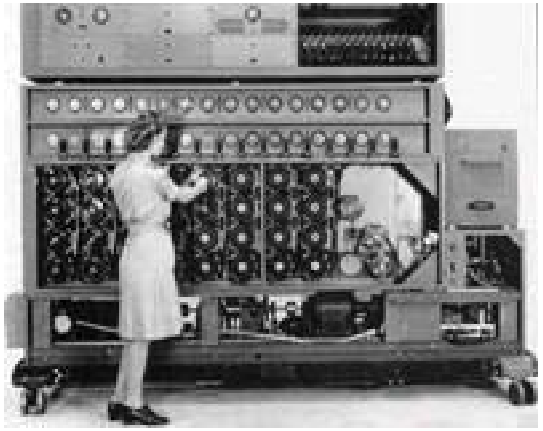
1842 - Ada Lovelace writes the first computer program


The Analytical Engine
1

bbage

WWII & Code Breaking

1941 - the first 'Bombe' is completed

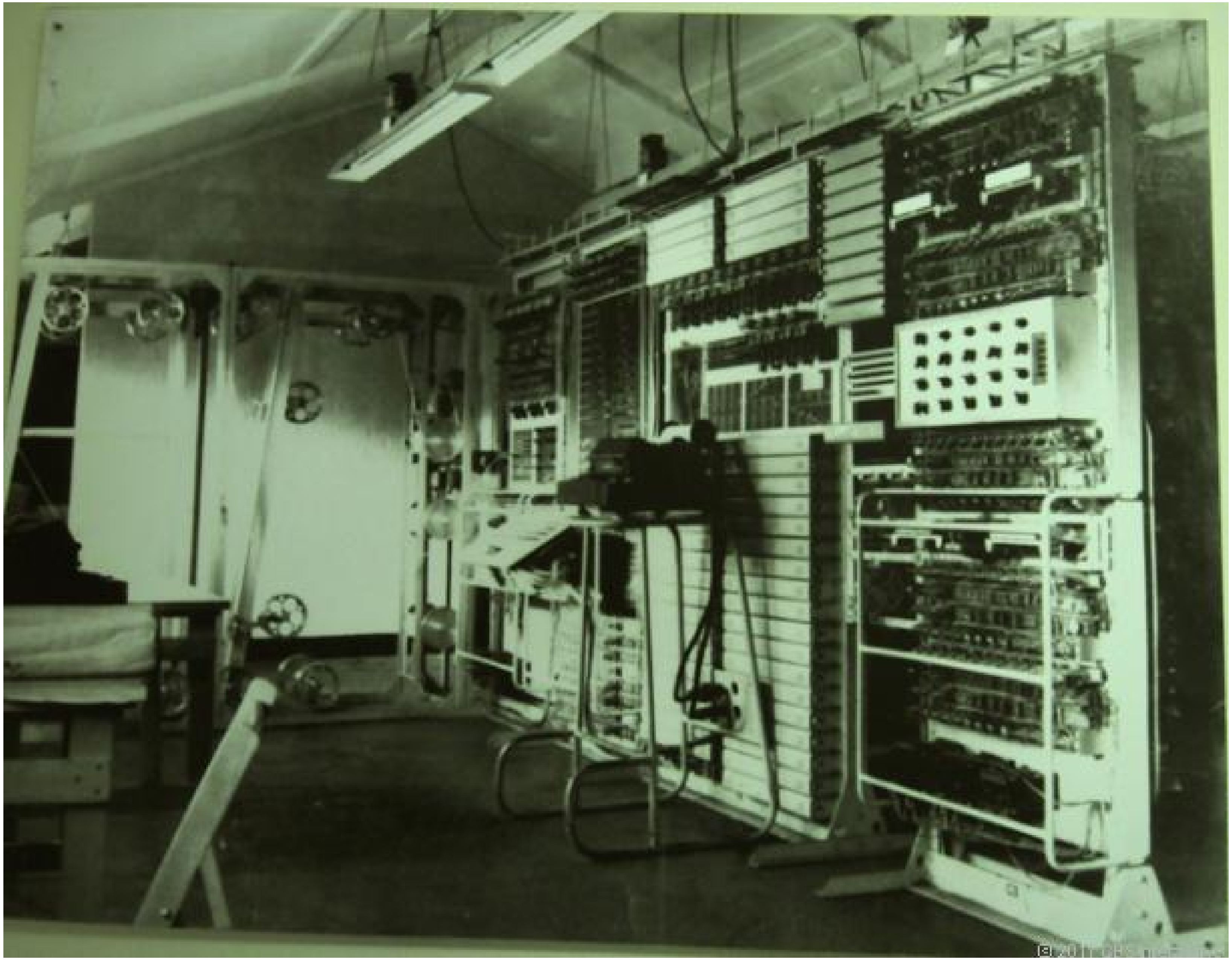


1942 - the Atanasoff-Berry machine built at Iowa State college

1944 - Colossus at Bletchley Park & the Enigma & Lorenz ciphers

1945 - John von Neumann wrote 'First Draft of a Report on the EDVAC'

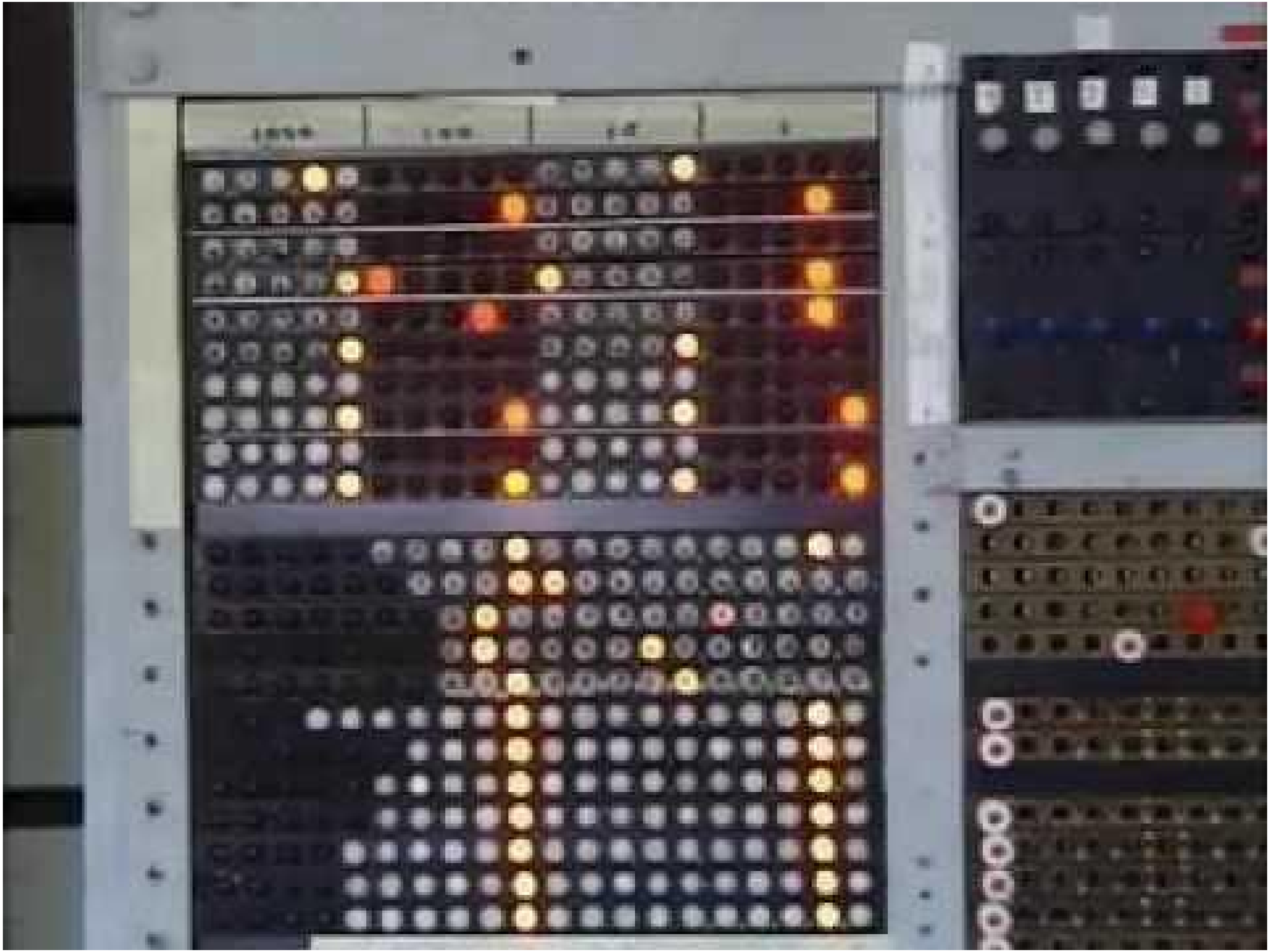
1946 - ENIAC announced to the public



© 2011

WWII & Code Breaking

- BBC article on Colossus code breaking of German signals



Computing

True or false?

- The majority of today's, and tomorrow's, most exciting and important arts, sciences, and technologies are driven by computing.

- A better understanding of computing helps illuminate insights and questions into the very nature of our minds, our culture, and our universe.

Processes

- what are processes?

- what are processes in the real world?

- computer science is often described as the study of **information processes**

- why abstract and not real/physical?

Procedures

- what is a procedure, and its relationship to a process?

- what is a mechanical procedure?

- what is an algorithm?

- example algorithm from the real world?

Algorithm

or How to make a single cup of Tea

- place tea bag into cup
- add water to kettle and boil
- pour boiling water from kettle into cup and over tea bag
- leave tea to diffuse in cup
- remove tea bag from cup
- add milk and sugar as required

Programming a computer

- requires tools that allow us to describe processes

- these processes need to be precise and succinct

- every step needs to be described

Therefore, what is a computer

In essence,

- Input

- Execution

- Output

"A computer terminal is not some clunky old television with a typewriter in front of it. It is an interface where the mind and body can connect with the universe and move bits of it about."

Douglas Adam

Therefore, what is programming?

- in its simplest form, programming is the creation of a series of instructions for the computer to follow, again step by step

- important to be clear and know what you want to program first

- identify the problem you want to solve

- in essence, programming can be seen as nothing more than problem solving

- there's no single best 'algorithm' for writing a software program

- effectively, when you program you are simply telling the computer what to do