PH345 Philosophy of Computation
Term 2, 2015-2016

- **Module tutor**
  - Walter Dean (W.H.Dean@warwick.ac.uk)
  - Put "PH345" in the subject line of your email and sign your *full name*.

- **Website**

http://go.warwick.ac.uk/ph345 (See "Course Materials" at the bottom.)
Readings, announcements, and problem sets will be posted at this address.

- **Logistics**
  - Lecture: Monday 12:00-14:00 in S0.19
  - Seminar: Monday 18:00-19:00 in MS.04

- **Related module**

PH340: Logic III will cover related material on Gödel's incompleteness theorems and undecidability.

- **Description**

The purpose of this module is to provide an accessible introduction to theoretical computer science and related philosophical issues about computation. Among the questions we will address are the following: What is a *model of computation*? Is it possible to provide a mathematical definition of the intuitive notion of a function *computable by an algorithm*? Do there exist functions which are *non-computable* (even in principle)? What does it mean to say that one function is *harder* to compute than another? What is a *complexity class* and what is the significance of the $\mathbf{P} \neq_? \mathbf{NP}$ problem? What is the relationship between computation and proof in mathematics?

- **Assessment**

Assessment for this module will consist of two components:

1) Assessed problem sets [15%]. There will be three problem sets posted on the module website. These are to be submitted at the beginning of seminar (= Monday 18:00) weeks 5 and 8 and to either to my pigeonhole in the department or by email in week 11.[1] Solutions will be posted on the module website.
2) One hour exam [85%]. This will be problem-based and administered during the summer exam period.

- **Seminar**

All students are expected to attend and participate in seminar. This is particularly important for Philosophy students as attendance information is collected on Tabula and feeds in to the monitoring point system. Seminar is your opportunity to clarify any issues from lecture or the readings and to get help with problems.

- **Outline and resources**

We will cover three general topics: *models of computation* (weeks 1-3), *computability theory* (weeks 4-7), *complexity theory* (weeks 8-10).

[1] Extensions can only be granted by the Senior Tutor for Philosophy upon submissions of evidence of extenuating circumstances.

Reading will primarily be drawn from the following sources:

- *Computability and logic* 5th ed. by George Boolos, John Burgess, and Richard Jeffrey, Cambridge University Press, 2007.
- *The Open Logic Text* (a part of the Open Logic Project).
- *Computability* by Nigel Cutland, Cambridge University Press, 1980.
- Computational complexity theory in the *Stanford Encyclopedia of Philosophy*
- *Introduction to the theory of computation* by Michael Sipser, Thomson, 2006.

I'll be using the following software packages for demonstrations:

- Tuatara Turing Machine Simulator (available as an executable `Java` file)
- DrRacket (a variant of `Scheme`)

You are encouraged to install and experiment with these yourself.

- **Approximate schedule and readings** (-) = required, (+) = recommended
  - **Week 1**: Introduction, the informal notion of algorithm, Turing machines
    (-) Boolos chapter 3
    (+) Turing machines [SEP]
  - **Week 2**: More Turing machines, register machines
    (-) Boolos chapter 3
    (-) Cutland Chapter 1
  - **Week 3**: Recursive functions, Church's Thesis
    (-) Open Logic "Recursive functions"
    (-) Cutland chapter 2-3
    (+) Recursive functions [SEP]
  - **Week 4**: Enumerating machines, universality, the Halting Problem
    (-) Open Logic "Computability theory" sections 1-9
    (+) Cutland chapters 4-5
  - **Week 5**: Recursive and recursively enumerable sets
    (-) Open Logic "Computability theory" sections 9-13
    (+) Cutland chapter 7
    (⋆) **Problem set 1 due**
  - **Week 6**: Reading week (NO LECTURE OR SEMINAR)
  - **Week 7**: Reducibility and completeness
    (-) OL "Computability theory" sections 14-19
    (+) Cutland chapter 9
  - **Week 8**: Measuring time and space complexity, the complexity class **P**
    (-) Computational complexity theory [SEP] sections 1-2
    (+) Sipser chapter 7.1-7.2
    (⋆) **Problem set 2 due**
  - **Week 9**: Nondeterminism, the complexity class **NP**, **NP**-completeness
    (-) Computational complexity theory [SEP] section 3
    (+) Sipser chapter 3.2, 7.3-7.5
  - **Week 10**: The **P** $\neq_?$ **NP** problem, mathematical proof and complexity theory
    (-) Computational complexity theory [SEP] section 4
    (+) Sipser chapter 10.1, 10.2, 10.4
  - **Week 11** End of term.
    (⋆) **Problem set 3 due**