

Curriculum Workshop

Department of Computer Science University of Cape Town

23 January 2017



Agenda

- Finalising the Honours 2017 calendar [Michelle]
- Redesigning Honours and Masters for 2018 and beyond [Hussein, Tommie, Michelle]
- Implications of the ACM Computer Science 2013 curriculum
- Decolonisation/Relevance interventions [Gary, Stephan]
- Evaluating our quality/plagiarism controls
- Milestones in our programme (e.g., the CS2 prac test)
- Soft skills (reading, writing, speaking, collaborating) across the curriculum
- Projects and Innovation across the curriculum
- Labs vs laptops what equipment do our students need?



Finalising the 2017 Honours Calendar



Michelle

Honours and Masters 2018+



Hussein, Michelle, Tommie

Motivation

- Our postgraduate programmes have not changed in form in 15 years.
- Our student cohort has changed significantly: more PhDs and more applications.
- But we are still stuck supervising 4-5 year MSc students.
- Is it time to move to a more EU/US model?
 - Coursework + Dissertation for MSc
 - Dissertation for PhD



Advantages of Coursework Masters

- Focus on research and research training as department's priority.
- Greater throughput of students.
- Deal with dwindling number of local MSc students and transformation of student body.
- Possibilities for part-time studies, specialised degrees and inter-department degrees.



Design Principles

Honours

- More like 4th year CS in other countries.
- All students should take Honours!
- Honours should take less departmental effort.
 Fewer options for students.
- Masters
 - Structured research training for ALL MSc students.
 - Larger class of students.
 - Shared model with Data Science and others.
 - Part-time and specialised degrees.



Honours 2018

Compulsory modules (64 credits):

- 8c RM
- 8c PCU
- 12c/15l Compilers 1
- 12c/15l Compilers 2
- 12c/15l Functional Programming
- 12c/15l Concurrency? (should we remove it from CS2)?
- Project (60 credits)



Honours 2018

Electives (36 credits):

- 3 modules of 12 credits each, for a total of 36 credits
- 12c/15l AI 1
- 12c/15l Graphics 2?
- 12c/15l Security
- 12c/15l Games?
- Students could take electives from Masters courses as well



Masters 2019 (or 2018?)

- compulsory (18 credits):
 - 18c RM
- electives (72 credits):
 - 6 modules of 12 credits (15I) each, for a total of 84 credits
 - IR / Visualization (shared with Data Science) / AI 2 / GPU / Statistics for HCI / Co-design? / Rural networking / Distributed Computing / guest courses like Nick's and Alastair's
 - students could take selected electives from Data Science (e.g., Unsupervised learning), MIT (e.g., Ethics)





Part-time and PGDip/Masters

- All MSc modules offered in afternoon to allow for part-time students.
 - Take coursework over 2 years, like MIT.
- Admission to PGDip based on 4th year performance.
- Upgrade to Masters on successful completion of coursework and funding a supervisor.



Workload

New (Honours + Masters) = more lectures

- New (Honours + Masters) Games = many fewer lectures
- New (Honours + Masters) Games + New CS3 = slightly fewer or same lectures





- UNIVERSITY OF CAPE TOWN
 - Games + New CS3
- No more Games major
- □ 4 courses in CS3:
 - CSC3002: OS + Net
 - CSC3003: SE + ToA
 - CSC3022: C++ + ML
 - CSC3024: Graphics + Web Technology
- Why? A fourth CS course keeps the most important element from Games and allows single major degrees!





Discussion



ACM CS Curriculum 2013





UNIVERSITY OF CAPE TOWN

Major changes since CS2008

- New knowledge areas
 - Information Assurance and Security (IAS)
 - Networking and Communication (NC)
 - Platform-Based Development (PBD)
 - Parallel and Distributed Computing (PD)
 - Software Development Fundamentals (SDF)
 - Systems Fundamentals (SF)
- 3 designation system
 - Core-1 = All topics must be covered.
 - Core-2 = Vast majority (80%) must be covered.
 - Elective = Some topics must be covered to provide a cohesive degree.



Algorithms

AL. Algorithms and Complexity (19 Core-Tier1 hours, 9 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
AL/Basic Analysis	2	2	Ν
AL/Algorithmic Strategies	5	1	Ν
AL/Fundamental Data Structures and Algorithms	9	3	N
AL/Basic Automata, Computability and Complexity	3	3	N
AL/Advanced Computational Complexity			Y
AL/Advanced Automata Theory and Computability			Y
AL/Advanced Data Structures, Algorithms, and Analysis			Y





Architecture and Organisation

AR. Architecture and Organization (0 Core-Tier1 hours, 16 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 Hours	Includes Elective
AR/Digital Logic and Digital Systems		3	Ν
AR/Machine Level Representation of Data		3	Ν
AR/Assembly Level Machine Organization		6	Ν
AR/Memory System Organization and Architecture		3	Ν
AR/Interfacing and Communication		1	N
AR/Functional Organization			Y
AR/Multiprocessing and Alternative Architectures			Y
AR/Performance Enhancements			Y





Computational Science

CN. Computational Science (1 Core-Tier1 hours, 0 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
CN/Introduction to Modeling and Simulation	1		Ν
CN/Modeling and Simulation			Y
CN/Processing			Y
CN/Interactive Visualization			Y
CN/Data, Information, and Knowledge			Y
CN/Numerical Analysis			Y





Discrete Structures

DS. Discrete Structures (37 Core-Tier1 hours, 4 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
DS/Sets, Relations, and Functions	4		Ν
DS/Basic Logic	9		Ν
DS/Proof Techniques	10	1	Ν
DS/Basics of Counting	5		Ν
DS/Graphs and Trees	3	1	Ν
DS/Discrete Probability	6	2	Ν





Graphics and Visualisation

GV. Graphics and Visualization (2 Core-Tier1 hours, 1 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
GV/Fundamental Concepts	2	1	Y
GV/Basic Rendering			Y
GV/Geometric Modeling			Y
GV/Advanced Rendering			Y
GV/Computer Animation			Y
GV/Visualization			Y





HCI

HCI: Human Computer Interaction (4 Core-Tier1 hours, 4 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
HCI/Foundations	4		N
HCI/Designing Interaction		4	N
HCI/Programming Interactive Systems			Y
HCI/User-Centered Design & Testing			Y
HCI/New Interactive Technologies			Y
HCI/Collaboration & Communication			Y
HCI/Statistical Methods for HCI			Y
HCI/Human Factors & Security			Y
HCI/Design-Oriented HCI			Y
HCI/Mixed, Augmented and Virtual Reality			Y





Information Assurance and Security

IAS. Information Assurance and Security (3 Core-Tier1 hours, 6 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
IAS/Foundational Concepts in Security	1		Ν
IAS/Principles of Secure Design	1	1	Ν
IAS/Defensive Programming	1	1	Y
IAS/Threats and Attacks		1	N
IAS/Network Security		2	Y
IAS/Cryptography		1	N
IAS/Web Security			Y
IAS/Platform Security			Y
IAS/Security Policy and Governance			Y
IAS/Digital Forensics			Y
IAS/Secure Software Engineering			Y





Information Management

IM. Information Management (1 Core-Tier1 hour; 9 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
IM/Information Management Concepts	1	2	Ν
IM/Database Systems		3	Y
IM/Data Modeling		4	Ν
IM/Indexing			Y
IM/Relational Databases			Y
IM/Query Languages			Υ
IM/Transaction Processing			Y
IM/Distributed Databases			Y
IM/Physical Database Design			Y
IM/Data Mining			Y
IM/Information Storage And Retrieval			Y
IM/MultiMedia Systems			Y





Intelligent Systems

IS. Intelligent Systems (10 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
IS/Fundamental Issues		1	Y
IS/Basic Search Strategies		4	N
IS/Basic Knowledge Representation and Reasoning		3	Ν
IS/Basic Machine Learning		2	N
IS/Advanced Search			Y
IS/Advanced Representation and Reasoning			Y
IS/Reasoning Under Uncertainty			Y
IS/Agents			Y
IS/Natural Language Processing			Y
IS/Advanced Machine Learning			Y
IS/Robotics			Y
IS/Perception and Computer Vision			Y

department of computer science



Networking and Communication

NC. Networking and Communication (3 Core-Tier1 hours, 7 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
NC/Introduction	1.5		Ν
NC/Networked Applications	1.5		Ν
NC/Reliable Data Delivery		2	Ν
NC/Routing And Forwarding		1.5	Ν
NC/Local Area Networks		1.5	Ν
NC/Resource Allocation		1	Ν
NC/Mobility		1	Ν
NC/Social Networking			Y





Operating Systems

OS. Operating Systems (4 Core-Tier1 hours; 11 Core Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
OS/Overview of Operating Systems	2		Ν
OS/Operating System Principles	2		Ν
OS/Concurrency		3	N
OS/Scheduling and Dispatch		3	N
OS/Memory Management		3	N
OS/Security and Protection		2	N
OS/Virtual Machines			Y
OS/Device Management			Y
OS/File Systems			Y
OS/Real Time and Embedded Systems			Y
OS/Fault Tolerance			Y
OS/System Performance Evaluation			Y





Platform-Based Development

PBD. Platform-Based Development (Elective)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
PBD/Introduction			Y
PBD/Web Platforms			Y
PBD/Mobile Platforms			Y
PBD/Industrial Platforms			Y
PBD/Game Platforms			Y



SPES DOMA

Parallel and Distributed Computing

PD. Parallel and Distributed Computing (5 Core-Tier1 hours, 10 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
PD/Parallelism Fundamentals	2		Ν
PD/Parallel Decomposition	1	3	Ν
PD/Communication and Coordination	1	3	Y
PD/Parallel Algorithms, Analysis, and Programming		3	Y
PD/Parallel Architecture	1	1	Y
PD/Parallel Performance			Y
PD/Distributed Systems			Y
PD/Cloud Computing			Y
PD/Formal Models and Semantics			Y



department of computer science



Programming Languages

PL. Programming Languages (8 Core-Tier1 hours, 20 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
PL/Object-Oriented Programming	4	6	N
PL/Functional Programming	3	4	N
PL/Event-Driven and Reactive Programming		2	N
PL/Basic Type Systems	1	4	N
PL/Program Representation		1	N
PL/Language Translation and Execution		3	N
PL/Syntax Analysis			Y
PL/Compiler Semantic Analysis			Y
PL/Code Generation			Y
PL/Runtime Systems			Y
PL/Static Analysis			Y
PL/Advanced Programming Constructs			Y
PL/Concurrency and Parallelism			Y
PL/Type Systems			Y
PL/Formal Semantics			Y
PL/Language Pragmatics			Y
PL/Logic Programming			Y



department of computer science



Software Development Fundamentals

SDF. Software Development Fundamentals (43 Core-Tier1 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
SDF/Algorithms and Design	11		N
SDF/Fundamental Programming Concepts	10		N
SDF/Fundamental Data Structures	12		N
SDF/Development Methods	10		N





Software Engineering

SE. Software Engineering (6 Core-Tier1 hours; 21 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
SE/Software Processes	2	1	Y
SE/Software Project Management		2	Y
SE/Tools and Environments		2	N
SE/Requirements Engineering	1	3	Υ
SE/Software Design	3	5	Y
SE/Software Construction		2	Y
SE/Software Verification and Validation		4	Y
SE/Software Evolution		2	Y
SE/Software Reliability		1	Y
SE/Formal Methods			Y





Systems Fundamentals

SF. Systems Fundamentals. [18 Core-Tier1 hours, 9 Core-Tier2 hours]

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
SF/Computational Paradigms	3		Ν
SF/Cross-Layer Communications	3		Ν
SF/State and State Machines	6		Ν
SF/Parallelism	3		Ν
SF/Evaluation	3		Ν
SF/Resource Allocation and Scheduling		2	Ν
SF/Proximity		3	N
SF/Virtualization and Isolation		2	Ν
SF/Reliability through Redundancy		2	Ν
SF/Quantitative Evaluation			Υ





Social Issues and Professional Practice

SP. Social Issues and Professional Practice. [11 Core-Tier1 hours, 5 Core-Tier2 hours]

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
SP/Social Context	1	2	N
SP/Analytical Tools	2		N
SP/Professional Ethics	2	2	N
SP/Intellectual Property	2		Y
SP/Privacy and Civil Liberties	2		Y
SP/Professional Communication	1		Y
SP/Sustainability	1	1	Y
SP/History			Y
SP/Economies of Computing			Y
SP/Security Policies, Laws and Computer Crimes			Y





Discussion



Decolonisation



Gary, Stephan

Quality / Plagiarism



Plagiarism

- No consistent application of plagiarism checking in different courses in different years.
 Too much academic time to police system?
- Need departmental level plan:
 - What is done at first year?
 - What is done at second year?
 - etc.





Plagiarism Checking Proposal

- **CS1**:
 - Randomly check 50% of assignments.
- **CS2**:
- Randomly check 2 assignments each semester.
 CS3+:
 - Randomly check 1 assignment each semester.



Quality Controls

How do we ensure spreadsheets are correct?

November 2016 exam had some spreadsheet errors caught after checking by lectures+cc+moderator+external!

How do we ensure quality of assessment?

- Some tests were too easy and some too difficult.
- External examiner commented on exam questions at higher levels.
- Bloom's taxonomy?





Discussion



Milestones and Projects



Degree Milestones

Current Status

- CS1: none
- CS2: Unix test, Java test
- CS3: SE Project
- CS4: Research Project

2 Types of milestones:

- Competency test
- Project



Issues

How are milestones implemented?

- Currently, they are associated with some courses as a DP requirement.
- How many attempts do students get?
 - Currently, varies based on pass/fail levels.
- How are achievements noted?
 - Currently, they are embedded in course results.
 - Some disgruntled students pass competency tests but have to repeat them.



Discussion Topics

- Can we have projects in CS1+CS2, marked by tutors?
- Can we separate milestones from courses?
 - Maybe using project course codes like GEO and MCB?
- Do we need other competency tests?
- How can we improve the programming competency test?



Soft Skills and Innovation



Issues

State of Play:

- Students are not reading until CS4.
- Scientific writing seriously taught in CS4.
 Compare to other scientific disciplines!
- Very poor report writing in CS1-3.
- Presentation only introduced in CS4.
- Innovation only introduced in CS4.
- Problems:
 - Soft skills are only taught in CS4.
 - Many students try to get these skills elsewhere.



Discussion Topics

- Can we require reading and writing in all courses?
 - Report in scientific format for every assignment?
 - Scientific evaluation of software built where appropriate?
- Do we require more presentation?
- Do we "teach" entreneurship and innovation across the curriculum?
 - Internships? Competitions? Special lectures?



Labs vs. Laptops



Discussion Topics

- We are already moving towards a laptop requirement from 2017 for CS2.
- How do we use the laptop requirement in evolving teaching?
- How does this influence practical tests? How we still need labs?
- What assumptions can we make about students and laptops?
- How do we distribute donated laptops?



Proposal: Labs

- CS4 lab is already mostly laptop space.
- CS2/3 lab should move slowly to become mostly laptop space.
 - At most 50% desktops by 2019.
- Games lab to be incorporated into Senior Lab.
- Senior lab Annex (aka TSL) to be fixed desktops (usable for controlled experiments at higher levels).



Proposal: Laptops

- Students to install Unix in virtual machines from 2017.
- Competency tests to be conducted in flat venues with laptops, using virtualization.
- Criteria for loaning donated laptops for semester:
 - first cohort: all students on NSFAS
 - second cohort: "missing middle"
 - third cohort: anyone else (but keep spares)
 - university can provide categories?



that's all folks!

