CITS1200 Java Programming
Semester 2, 2011

The School of Computer Science & Software Engineering
The University of Western Australia

Unit Coordinator: Prof. Amitava Datta
Tutor: Miss Ariel Xiaohang Ma
CITS1200 Overview

• The aims of CITS1200 are that students learn
  – the object-oriented programming paradigm (OO or OOP)
  – how to design simple programs in this paradigm
  – how to implement these programs in the Java language

Computer Science is no more about computers than Astronomy is about telescopes. (Edsger Dijkstra)
CITS1200 Overview

• CITS1200 is a first programming unit:
  – no specific programming knowledge is assumed
  – there is a wide range of student backgrounds
  – it leads into CITS2200 Data Structures and Algorithms

• CITS1200 is the first in a series of units designed to give you
  – good problem-solving skills
  – a thorough understanding of the programming process
  – skills in several important modern languages and paradigms

No-one can turn you into a skilful and productive programmer who is employable in the software industry in three months.
CITS1200 Philosophy

- Core, foundation unit for all CSSE majors and degrees
- Strong emphasis on *fundamentals* of object oriented programming
  - objects, classes, and methods
- Assessment is based on *practical programming tasks*
  - rote learning is not enough
- Designed to
  - prepare students for CSSE majors
  - expose students to the science/engineering/art/craft of programming
  - encourage students to think independently at an abstract level

*Learning to program is like learning to swim or learning to juggle – you cannot get better by watching, only by doing!*
People & Places

• CIT1200 coordinator: Prof. Amitava Datta
  Tutor: Miss Ariel Xiaohang Ma
  – consultation times: Anytime or 2-3 on Fridays

• Lectures
  – GGGL:WOOL LT, 1-2 pm on Mondays
  – GGGL:WOOL LT, 1-2 pm on Thursdays
  – recorded with Lectopia, both audio and screen capture
  – you should buy a bound set of notes from the Co-op bookshop ($8.65)

• Labs
  – supervised lab sessions run from Week 2 in Lab 2.05
  – An extra lab has been added Friday 2-5 pm
  – you need not attend if you have completed the work, e.g. at home
  – you are encouraged to use your own laptop – wireless access is provided free
Java Workshop

- GGGL:WOOL LT, 2-3pm on Mondays from Week 2
- **Aimed at students with no prior programming experience who are having difficulty with the content or pace of the material**
- Provides an environment for
  - more leisurely explanation/exploration of fundamental concepts
  - additional examples complementing those in the lectures
  - embarrassment-free zone for *any questions*, no matter how elementary
- Purely optional on an as-needed basis
  - all are welcome, but the pace is slow, and sophisticated/advanced questions are not appropriate in this forum
  - come when you feel you need additional help with the lecture material

*The only stupid question is the one you didn’t ask!*
Java Workshop

- The exception to the last slide is that the workshop in Week 2 will be an introduction to the development tools that you will use in CITS1200
  - The interpreter
  - The editor
  - The testing environment
  - Installing these at home

*Everyone should attend the workshop in Week 2*
**CITS1200 Information**

- **Website**
  - *everything* that is distributed in CITS1200 will be available from

- **Discussion list**
  - uses the *cshelp* facility
  - to make a query, or to view or post answers, go to
Discussion forum for class questions, based on the philosophy of

**READ FIRST: if the answer is not there, THEN POST.**

- **Asking** questions is useful
  - formulating a question helps you to focus on the issue and to clarify what you understand and what you don’t, and helps others with similar problems
  - often, students find that writing down a question is enough on its own to resolve the issue

- **Answering** questions is useful
  - explaining something to somebody helps you to understand it more deeply, encourages you to put it into a suitable context, and helps both yourself and the student asking the question

- **Do not** post code to *help1200* if it is relevant to an assessment
CITS1200 Administration

- Administrative enquiries should go to
  
  amitava.datta@uwa.edu.au or, 20541009@student.uwa.edu.au

- Email us only regarding issues that are personal to yourself
  - illness or other personal issues
  - family issues
  - absences from class
  - requests for extensions

- For all other questions related to Java programming, use help1200
CITS1200 Announcements

• Announcements will be made in three places
  – in lectures
  – on the CITS1200 web-site
  – on help1200

When an announcement has been made in these three places, we will assume that you are aware of it.

• Any information distributed by email will be sent to your UWA or CSSE accounts
  – you should check your account regularly, or forward it
Recommended Books (1)

Object-Oriented Programming with Java: An Introduction
David Barnes

- We will cover Part One
  - 11 chapters
  - 328 pages
Recommended Books (2)

**Objects First With Java**
David Barnes & Michael Kölling

- BlueJ-aware text
- More abstract than *OOP with Java*
- Focuses on objects and classes, rather than lower-level constructs
Recommended Books (3)

How to Think Like a Computer Scientist (Java Version)
Allen B. Downey

- Free on-line book, available from the CITS1200 web-site
- Covers the material in a somewhat different order to CITS1200
CITS1200 Environment

- BlueJ integrated development environment
  - unique teaching environment
  - context sensitive editor
  - debugging environment
  - interactive object creation, manipulation, and inspection

CITS1200 Assessment

• Lab work
  – Weeks 4, 6, and 8
  – 2% + 3% + 5% = 10%

• In-class multiple-choice test
  – in the lecture on Thursday 15 September
  – 15%

• Programming Project
  – handed out in Week 9 and due end of Week 13
  – 15%

• Final Examination
  – 60%
Satisfactory Progress

• In order to pass CITS1200, consistent practice and effort are needed throughout the semester
• A small proportion of students make no real effort in the labs and early coursework, and inevitably fail the exam
• To address this, we define satisfactory progress to be

Achieving at least 40% (i.e. 10/25) for the graded lab sessions and midterm test (combined).

• If you do not make satisfactory progress, you may be excluded from the CITS1200 examination
• Excluded students will be notified in writing by Week 10
To pass CITS1200 (and to do well at UWA generally), you need to take responsibility for your own learning

- *attend* lectures regularly, preferably in person or at least via Lectopia
- *ask* questions on *help1200* or in labs, whenever you get stuck
- *use* the facilities provided by Student Services regarding study techniques, time-management, goal-setting, etc.
- *be aware* of managing your time early in the semester: you need to spend about ten hours/week over the entire semester on CITS1200
- *install* Java and BlueJ on your personal computer early in semester
- *practice, practice, practice*: programming is (at least partly) a skill, and you need to practice it to become proficient
If you run into problems…

• If you are struggling, the most important thing is to recognize it and to respond early: letting it slide is always ineffective
  – make sure you use all the help that is available

• Be realistic about your strengths, interests, and motivation
  – almost everyone who fails Java Programming does so not because of lack of ability, but because of lack of interest
  – the most common way to fail is to fall behind, start missing lectures and lab sessions, and then try to cram at the last moment
  – withdrawing is better than failing!

Failing a unit is expensive, it permanently mars your academic record, and it is almost always completely optional!
Academic Conduct Essentials (ACE)

- All newly enrolled students are required to complete a short compulsory online module called Academic Conduct Essentials (ACE), within the first ten weeks of semester
- ACE introduces students to essential knowledge regarding ethical scholarship, helps prepare them for the expectations of their university career, and informs them of correct academic conduct
- Non-completion within the required timeframe is documented on formal academic records
- More information on ACE is available at http://ace.uwa.edu.au
Plagiarism

- Plagiarism is “submitting for assessment work that was partially, or completely, done by someone else”
- Plagiarism is a major problem in all universities, particularly in CS and IT units
- Plagiarism is unfair to honest students, undermines the credibility of the course, and rarely helps the plagiarist in the long run, because they fail to acquire essential skills

All submitted lab work may be examined for similarities by software, and you may be asked to explain or reproduce your submitted work under controlled conditions.
Penalties for Plagiarism

- All instances of plagiarism, no matter how minor, are referred to the Academic Conduct Adviser (ACA), who determines penalties.
- A permanent record is kept of all instances of plagiarism, and this record is available to every ACA in the university.
- *No distinction* is made between the person who copied and the person who allowed their work to be copied.

*If your code is flagged by the software for similarities, but you are confident that you have not been involved in either deliberate or inadvertent plagiarism, come and see us immediately.*
**Scope of CITS1200**

- Fundamentals of OO programming
  - classes, objects, methods, interfaces, and encapsulation
  - NOT polymorphism or inheritance
- Design of classes
  - class identification, division of responsibilities
  - NOT modelling languages (UML)
- Implementation of classes
  - variables, data types, and expressions
  - control flow, repetition, iteration, and recursion
  - arrays, loops, elementary graphics
  - composition
  - NOT applets, GUIs, threads, files, I/O

*There are two ways of constructing a software design. One way is to make it so simple that there are obviously no deficiencies. And the other way is to make it so complicated that there are no obvious deficiencies. (Tony Hoare)*
Week 1 in the Labs

• Obtain your CSSE computer account name and password
  – you will need your student number and Pheme password
• Obtain your timetable from OLCR
• Activate your UWA email account
  – CSSE email will automatically be redirected to this account
• Go to Lab 2.03 or 2.05 and follow the instruction sheets
  https://secure.csse.uwa.edu.au/run/csentry?pw1=yes

• New students should also
  – familiarise themselves with the School web-site generally