ENGG1811 Computing for Engineers

Course Review and Exam Structure
Course objectives

From the Course Introduction:

• What you should be able to do by the end of the semester:
  
  – use spreadsheets and their associated tools to solve small computational problems in Engineering, Science and Business;
  
  – design and implement solutions to problems by writing small programs using a scripting language such as OpenOffice Basic or equivalent;
  
  – use a numerical computing environment such as MATLAB® to analyse, model and visualise data and systems; and
  
  – understand a bit about where the technology is and might be going (so you can be prepared to exploit it in your professional career)
How are these goals measured?

- **Laboratory work** – most skills developed here
- **Assignments** – more extensive exploration of some aspects, learning is assessed in final exam
- **Mid-session exam** – limited in scope and primarily for progress feedback
- **Final exam** – broad and intensive, determines final grade in most cases

<table>
<thead>
<tr>
<th>Labs (10)</th>
<th>Assignments (15)</th>
<th>Mid-semester Test (15)</th>
<th>Other (40)</th>
<th>Exam (60)</th>
<th>Final Mark (100)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Non-linear summation (see slide 5)</td>
<td>assignment-related (18) other (42)</td>
<td></td>
</tr>
</tbody>
</table>
Final Mark Calculation

- **If your exam is OK** (at least 20/60) you get the sum of the component marks

- **If your exam is not OK** (less than 20/65), then
  \[ \text{Final} = 3.75 \times \text{Exam} \times \text{Other} \div (\text{Exam} + \text{Other}) \]

Your mark can’t exceed 50 ...
Exam format

- **2 hours** (plus 10 minutes reading time)
- **Two kinds** of questions reflecting and testing different levels of understanding
  - Multiple choice (recalling and applying knowledge)
  - Programming exercises (applying knowledge and analysis skills)
- No supporting materials (but table of OO Basic functions and Matlab syntax provided)
- Paper isn’t released
- Sample paper is identical in structure
### Exam structure

<table>
<thead>
<tr>
<th>Sect.</th>
<th>Type</th>
<th>Answer Method</th>
<th>Questions, Marks</th>
<th>Approx. Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Multiple choice</td>
<td>Answer sheet</td>
<td>24 Q, 30 marks</td>
<td>30 minutes</td>
</tr>
<tr>
<td>B</td>
<td>OO Basic, written</td>
<td>White exam book*</td>
<td>2 Q, 30 marks</td>
<td>40 minutes</td>
</tr>
<tr>
<td>C</td>
<td>Matlab, written</td>
<td></td>
<td>3 Q, 40 marks</td>
<td>50 minutes</td>
</tr>
</tbody>
</table>

* *You can ask for an extra book if you run out of space*
Section A

- Each question has 4 choices
  - 1.25 marks for a correct answer
  - 0 marks for a missing or wrong answer
- Erase any old answer completely
- Double-check that your student ID is clearly shaded in the appropriate boxes and is 100% correct
- Double-check that your name is clearly written and shaded
- This section is scanned, not human-marked

Topics (approx.):
- 8 questions on OO Calc
- 8 questions on OO Basic
- 8 questions on Matlab

Remember to bring a soft (2B) pencil and eraser
Programming questions

• Section B: OO Basic
  – Two questions: 2 x 15 marks
  – B1 related to assignment

• Section C: Matlab
  – Three questions: 2 x 15 marks, 1 x 10 marks
  – C1 related to assignment
Answering programming questions

- Think about the problem first before you put the answer down

- OO Basic/Matlab details are not critical, you are being assessed on your ability to devise algorithms and express them in a formal notation

- Sketch out your solution first, present (relevant!) pseudocode if you are running out of time

- Comments are helpful for us to understand what you want to do

- In decreasing order of acceptability/marks earned for providing your solution
  - OO Basic / Matlab code (most acceptable)
  - Pseudo code
  - English sentences (least acceptable)
Answering programming questions (cont’d)

- **OO Basic:** Do “dim” the variables so we know the data type

- **Syntax error**
  - Spelling errors that do not caused ambiguity will be ignored
  - Syntax errors that do not caused ambiguity in interpreting your algorithm will be ignored
    - Example: If you write “End” instead of “End If” in OO Basic
    - Example: Next slide

- **Style:**
  - Good variable names help us to understand what you do
  - Indentation not marked but make it easier to read your code
  - Otherwise not assessed
Syntax errors that can cause ambiguity

- Must indicate the end of loops/if
  - Don’t forget End If, Wend, Next (in OO Basic)
  - Don’t forget end in Matlab

- Syntax in cheat sheets provided

---

**Unambiguous code**

```plaintext
sum = 0;
for j = 1:10
    sum = sum + j;
end
sum = sum + 10;
```

**Ambiguous code: Can’t tell which statement(s) is(are) under the for loop**

```plaintext
sum = 0;
for j = 1:10
    sum = sum + j;
sum = sum + 10;
```
General exam approach

- Use reading time to plan (also for writing name etc on answer sheet and books)
- Plan your use of the available time!
- Answer straightforward questions first
- Watch the time!
- If you’re hoping for just a **pass**, concentrate on using the time to produce **reliable answers** to a majority of questions, not necessarily all of them
- Watch the time! Unfortunately, "*I ran out of time*" doesn’t get you any sympathy marks
How to prepare

Lecture notes
• Review and properly understand the material
• Concentrate on examples (mainly Basic, Matlab)

Lab exercises
• Attempt any exercise you didn’t quite get to
• First exercise is generally most relevant (week 6 onwards)
• Lab 13 is especially valuable practice!

Sample exam
• Attempt under exam-type conditions if possible
• Solutions available a week or so before exam

• Forum and consultations
  – I will provide extra consultations before exam. Schedule will be posted on the notice page.
Give aways

• Don’t have to memorise precedence, will be given
• There are two cheat sheets OO Basic / Matlab. They are the same as those in the sample exam paper
• No questions on:
  – Week 11A: Hard Problems
  – Week 12A: Machine Learning
What if I’m sick?

- Only attend the exam if you are **functioning** fairly **normally**
- Get a **medical certificate** on the day
- **Submit** Application for Special Consideration to Student Central
- **Supplementary** assessment is **not automatic**, depends on reasonable effort in other parts of the assessment

**Supplementary Exam**

- Time/Date: ??? (check with the school office)
  - Location: To be advised later
- Please **reconfirm** the **time, date** and **location** of the supplementary exam by visiting the class web page after the final exam and just before the above date.
- There will be **NO further supplementary** dates.
- Similar format, but may not have choice in Part B and C
- No additional consultation or help
Last words

• Please complete CATEI survey
  – Good/bad/more of this/less of that/what can be done better

• Tutoring opportunities for ENGG1811

• Three parting messages:
  – Computing and programming are very useful in solving many engineering problems.
  – Learning programming is like learning a new language. If you practice, you will get better. Explore and learn more if you can
  – This world wants people with multiple skills. Important to find your talents and passions, but try also to explore and learn as many different areas as you can.