INTRODUCTORY PROGRAMMING

700204/900084

2018

UNIT OUTLINE
<table>
<thead>
<tr>
<th><strong>Unit name</strong></th>
<th>Introductory Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Unit number</strong></td>
<td>700204/900084</td>
</tr>
<tr>
<td><strong>Coordinator</strong></td>
<td>Zdenka Misanovic</td>
</tr>
<tr>
<td><strong>Session</strong></td>
<td>2018.3</td>
</tr>
</tbody>
</table>

**Handbook summary**
This unit introduces students to the principles required for the effective design and development of computer programs. This unit has been developed to help students acquire an understanding of essentials in designing programs theoretically and implementing them practically, using an integrated development environment (IDE).

**Credit point value**
10

**Prerequisite/s**
N/A

**Corequisite/s**
N/A

**Unit incompatible with and not to be counted for credit with**
N/A

**Assumed knowledge**
The ability to create a mathematical expression for a given problem scenario. This would require knowledge of basic arithmetic, percentages and simple statistical measures.

**Unit level**
Level Z — Non-award unit

**Attendance requirements**
Students are expected to attend at least 80% of classes. Educational research consistently demonstrates that this attendance level is associated with a high likelihood of achieving a passing grade.

**Enrolment restrictions**
Students must be enrolled at The College.

**Learning outcomes**
On successful completion of this unit, students should be able to:
1. define terms such as structured programming, variables, constants, control structures, modularisation, cohesion, coupling, function, procedures, parameters and arguments
2. illustrate the steps involved in program development
3. solve problems and illustrate solutions using sequence control structure flowcharts and/or pseudocode, which are then coded in a 3GL language such as C++
4. solve problems and illustrate solutions using selection control structure flowcharts, pseudocode and translate to C++
5. solve problems and illustrate solutions using iteration control structure flowcharts, pseudocode and translate to C++
6. solve problems using modularisation with parameter passing
7. code, debug and test programs in C++ using an Integrated Development Environment (IDE), and
8. develop a set of input test data and desk check pseudocode.

Unit content

In this unit students will learn about:

- introduction to steps in program design and development
- problem solving and developing algorithms using IPO charts, flow charts and pseudocode
- sequence control structure
- selection control structures using IF and CASE statements
- repetition control structures using REPEAT, WHILE and FOR loops
- modularisation, procedures/void functions, functions and parameter passing
- input output data tables and desk check tables
- use of an Integrated Development Environment (IDE), and
- translating pseudocode into C++ using correct syntax.

Mode of delivery

This unit consists of six hours of supervised computer laboratory sessions. In addition, students will be required to access vUWS regularly, in order to download additional learning material, and to check for any announcements about the unit that may be posted there.

Online learning requirements

Essential resources

Essential texts
The College 2018, Introductory programming workbook, Western Sydney University The College, Sydney.

Further resources
Farrel, J 2013, Programming logic and design: comprehensive version, 7th edn, Course Technology, Boston, Mass.

Essential equipment

- A USB flash memory drive is strongly recommended for transporting files between home and The College.
**ASSESSMENT ITEMS AND WEIGHTING**

Assessment for this unit will be based on the following components:

<table>
<thead>
<tr>
<th>Task</th>
<th>Weighting</th>
<th>Learning outcomes assessed</th>
<th>Mandatory task</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Class work</td>
<td>20%</td>
<td>3–8</td>
<td>Yes</td>
</tr>
<tr>
<td>2. Class test (1.5 hours)</td>
<td>20%</td>
<td>3–5</td>
<td>Yes</td>
</tr>
<tr>
<td>3. Programming project (approx. 100 lines of code)</td>
<td>20%</td>
<td>3–8</td>
<td>Yes</td>
</tr>
<tr>
<td>4. Final exam (2 hours)</td>
<td>40%</td>
<td>1–8</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>100%</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For details of assessment due dates, please refer to the learning guide for this unit.

All marks will be determined in accordance with The College [Assessment Policy](#).

All assessment tasks are mandatory unless otherwise specified. Should a student fail to attempt/submit the first formal assessment task in a unit, they will be deemed to be at risk and will need to follow an intervention plan in order not to receive a Fail Non-Submission (FNS) grade. However, failure to attempt/submit all other mandatory assessment tasks will result in an immediate FNS grade for the unit.

In order to pass this unit, students must:

- attempt/submit all mandatory assessment tasks including the final exam, and
- obtain a minimum overall mark of 50%.