|  |
| --- |
|  |
|  |
| **School of Arts, Media and Computer Games** |

**Assessment Instrument Coversheet**

Module Code: AG0701a

Module Title: Programming in C++

Lecturer: Dr Natalie Coull and Dr Suzie Prior

Submission Date: Friday 4th December 2015 23:59hrs

Feedback Return Date: Friday 18th December 2015 17:00hrs

Feedback Type: verbal

(eg verbal, Blackboard)

Grading Criteria Refer to Page 4

**Submission Requirements:**

Your assessment must be submitted via Blackboard. The maximum file size which can be submitted is 20MB so you may need to reduce the size of any image files within your document.

Guidance on submitting via Blackboard is available at: <http://submit.ac.uk/en_gb/training/student-training/submitting-a-paper>, but please contact the Support Enquiry Zone on 01382 308833 or [sez@abertay.ac.uk](mailto:sez@abertay.ac.uk) if you have any problems with submitting your work on the Blackboard.

Submission of your work after the submission date deadline will be deemed as late submission and will incur penalty, including the possibility of the work being awarded a non-submission (NS) grade.

**AG0701A Programming in C++**

**1 Introduction**

Module AG0701A Programming in C++ is assessed solely by a portfolio of programs, which demonstrate your skills and evidence achievement across the module. The portfolio submission from semester 1 will form 50% of the overall module grade for this module.

**2 Learning Outcomes**

All modules at Abertay are defined by module descriptors which are published on Oasis. In each case the module seeks to encourage the student to achieve “Learning Outcomes” and this provides the focus, in particular, of assessments – you need to show that you have achieved them. For AG0701A these outcomes are:

1. Apply knowledge of C++ syntax in the construction of programs.
2. Apply pseudo-code, a modular, and an object-based approach to program construction.
3. Demonstrate the role of data representation, data structures, recursion and finite state machines in program design.
4. Determine the costs and/or benefits of different coding alternatives.

By the time you have completed the portfolio exercises across semester 1 and semester 2, you will have evidenced all of these outcomes. This portfolio directly relates to outcome 1, some of outcome 2 and outcome 3.

**3 The Portfolio**

The portfolio for semester 1 consists of three programs from the list of exercises below. You are required to submit a portfolio of work to reflect progress of the Learning Outcomes of the module. You need to choose three exercises and put them into your portfolio.

The exercises, on which the assessment for the module is based, are:

* Specific Modifications on Andrew LaMothe’s Game code (wk3)
* A Fruit Machine (wk4)
* Hotel booking system(wk5)
* Array sort (wk6)
* A Finite State Machine or Cyber Pet (wk9) (this is needed to achieve an A or B grade)

You must choose 1 program from weeks 5 and 6…and 2 programs from weeks 3,4 and 9

**3.1 Writing your programs**

Each of the three programs you select should include the following parts:

* A workable cleaned Visual Studio project with source code
* Some commentary in the sources code to explain each section of the code:
  + what it does,
  + how well it works,
  + any special features or bugs
* An executable of the program

**4 Submission**

**Part 1:** You need to submit one file to blackboard: a compressed zip file of the 3 programs, making sure that executables run directly without the need of the compiler environment.

The final date for work is **23:59hrs Friday, 4th December 2015.** For the submission rules, refer to the academic regulations.

**Part 2:** In addition to submitting evidence of your work via Blackboard, students are required to have their work ‘signed off’ by a tutor to evidence ownership and understanding of the code. This will be scheduled for the practical sessions of week 14 (i.e w/c 7th December 2015). Where possible, students who have completed the work in advance can get their programmes signed off earlier.

**4.1 How you will be assessed**

Your electronic submission to blackboard is used to demonstrate your achievements to the moderator and external examiner. Your module tutor will assess your work during the demonstration, where you will be asked some question on your code. These questions will be straight-forward and simply help us to determine that you understand the code. Examples of questions which might be used to determine understanding of Week 4 question 1 would be:

* What would you have to change in this code to allow the user to quit using 0 instead of -1
* How easy would it be to enable the user to quit using the word ‘quit’ instead of an integer

You will receive feedback on your programs during the demonstration, and the grades will be issued electronically either via email or via Blackboard once all students have completed their demonstrations. Students who have not uploaded their programs to blackboard will not be allowed to demonstrate them.

The grade will be calculated based on the number of programs completed and your ability to demonstrate an understanding of the code. Completing the finite state machine program from week 9 is essential to achieving an A grade.

**Grading Criteria**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Grade** | **C++ Knowledge** | **Programming** | | **Experiments report** | **Report writing and formatting** |
| A/A+ | An excellent grasp of knowledge and understanding of C++ syntax, program structure, Compound data types, recursion and finite state machines are evidenced | Excellent programming quality, following a consistent style and making use of generally accepted best practices during implementation. | | Excellently describes experiments to be implemented. Sets context for experiments, clear and detailed steps and presentation of results. | Excellent formatting and structure, with clear flow of information between sections. Excellent use of language with correct grammar and spelling. |
| B/B+ | A very good grasp of knowledge and understanding of C++ syntax, program structure, Compound data types, recursion and some aspects of finite state machines are evidenced | Very good programming quality, following a consistent style. | | Very good description of experiments to be implemented. Sets context for experiments, clear and detailed steps and presentation of results. | Very good formatting and structure, with good flow of information between sections. Excellent use of language with correct grammar and spelling. |
| C/C+ | A good level of knowledge and understanding of C++ syntax, program structure, Compound data types and recursion are evidenced | Good programming quality, with some minor inconsistencies. | | Good description of experiments to be implemented. Sets context for experiments, detailed steps and presentation of results. | Good formatting and structure with reasonable flow of information between sections. |
| DD+ | Satisfactory knowledge and understanding of C++ syntax, program structure, Compound data types and recursion are evidenced | Satisfactory programming quality, with substantial scope for improvement. | | Satisfactory description of experiments to be implemented. Sets context for experiments, steps and results. | Satisfactory formatting and structure but lacking coherence of sections. |
| MF | Marginally unsatisfactory knowledge and understanding of C++ syntax. | Marginally unsatisfactory programming quality. | | Marginally unsatisfactory description of experiments to be implemented. | Marginally unsatisfactory formatting and structure. |
| F | | | Performance well below the threshold level, with only limited evidence of achievement | | |
| NS | | | There is no submission, or the submission contains no relevant material | | |