Concordia University
Department of Computer Science
and Software Engineering

Advanced Program Design with C++
COMP 345 --- Fall 2015

Project Intermediate Build Grading

1. First Incremental Code Build Description

You must deliver an operational version demonstrating the full capacity of your system. This is about demonstrating that the code build is effectively aimed at solving specific project problems and completely implementing specific system features. The code build must not be just separated portions of the final project, but a fully operationally integrated software that can be demonstrated by its operational usage.

The presentation should be organized as follows:

1. Brief presentation of the Design, and Use of Tools as listed below under “Graduate attributes—skills**”**
2. Demonstration of the functional requirements as listed below under “Functional Requirements”.

You are graded according to how effectively you can demonstrate that the features are implemented. If you cannot really demonstrate the integrated features through execution, you will have to prove that the features are implemented by explaining how your code implements the features and what are the expected integration problems, in which case you may lose some marks, even if your explanations are satisfactory.

During your presentation, you have to demonstrate that you are well prepared for the presentation, and that you can easily provide clear explanations as questions are asked about your understanding of the problem being solved, the structure and functioning of your code, as well as your use of tools.

2. Team Identification

Team	Evaluator	Signature	Date	Time

3. Grading

Functional Requirements			**35**
Map creation and editing			**16**
User-driven creation of map elements, such as country, continent, and connectivity between countries.			5
Saving a map to a file exactly as edited.			3
Loading a map from an existing file, then editing the map.			3
Verification of map correctness before saving (at least 3 types of incorrect maps).			5
Game Play			**19**
Implementation of a game driver implementing the game phases.			4
Startup phase			**7**
Game starts by user selection of a user-saved map file, then loads the map as a connected graph.			5
User chooses the number of players, all countries are randomly assigned to players.			2
Reinforcement phase			**5**
Calculation of correct number of reinforcement armies.			2
User-driven placement of reinforcement armies on the map.			3
Fortification phase			**3**
Implementation of a valid fortification move according to the Risk rules.			3
Graduate attributes—skills			**15**
Knowledge-base	Indicator 1.3: Knowledge-base in a specific domain: demonstrated knowledge of programming principles used in the implementation.		1
Design	Indicator 4.1: Problem identification and information gathering: knowledge and correct understanding of the functional requirements and the game rules.		2
	Indicator 4.3: Architectural and detailed design: Rationale for overall project architectural structure. Demonstration/explanation of the correct use of three different design patterns such as those implemented in the individual assignments.		3
	Indicator 4.4: Implementation and validation: Correct use of C++ features leading to stable execution that has been properly tested in various situations.		2
Use of tools	Indicator 5.1: Ability to use appropriate tools, techniques and resources: proficient use of particular tools (C++ language, libraries, project management tools, etc.) for the implementation.		2
	Indicator 5.2: Ability to select appropriate tools, techniques, and resources: justified adoption of tools in the project (e.g. compiler, IDE, libraries, project management tools, etc).		1
Communication	Indicator 7.3: Documentation: Code readability: layout, naming. Consistent use of comments		2
	Indicator 7.4: Oral presentation: Structure and demonstrated preparation of presentation, using appropriate presentation techniques. Demonstrated knowledge of code base/clarity of explanations.		2
Total			**50**