

**Concordia University
Department of Computer Science
and Software Engineering**

**Advanced Program Design with C++
COMP 345 --- Fall 2013**

Project Build 1 Grading

1. First Incremental Code Build Description

You must deliver an operational version demonstrating some capacity of your system. This is about demonstrating that the code build is effectively aimed at solving a specific project problem or completely implementing specific system features. The goal of the build must be presented first, then demonstration is made to show that the goal has been met, and also to explain some parts of your code. The code build must not be just a "portion of the final project", but rather be something useful with a purpose on its own.

2. Team Identification

Team	Evaluator	Signature	Date	Time

3. Grading

Presentation		5
Start with a brief and clear explanation of the architectural design of the entire project		1
Knowledge of code base/clarity of explanations when questions are asked		2
Effectiveness and demonstrated preparation of the presentation		2
Character generation/edition		13
Creating/editing a <i>fighter</i> character following the d20 game rules: level, ability scores and modifiers, hit points, armor class, attack bonus, damage bonus		4
Inventory pane, including worn items slots (one of each: armor, hands, rings, helmet, boots, belt) and backpack (item container). Equip /unequip items, i.e. move items between worn item slots and the backpack.		5
Save/load a character to/from a file		4
Map generation/edition		14
Create/edit a map of a user-defined size		4
Place game elements on a map (entry point, exit point, doors, chests, monsters, ...)		5
Distinction between game elements on map display		1
Save/load a map to/from a file		4
Play		13
Select a map and character from a list of saved ones		2
Adapt the map elements (opponents, treasure) to the level of the character		3
Start the game by having the player character placed on the starting point		1
Move the character, square by square on the map		2
Toggle a view of character information (player or opponents) and chest content during play.		3
End the game by having the character stepping on the exit point		1
Make the character go up a level upon completion of the map		1
Design and programming style		5
Use of significant names for identifiers, code readability, comments, modular design		2
Use of observer pattern for character, map, and item containers (backpack/worn items, chest)		3
Total		50