**Grading Guidelines for Project 5**
**COP 3330 - Instructor Prof Ann Ford Tyson**
**Fall Term 2007**

Total Possible Points: 100
Project Due Date: Wednesday November 14, 2007

This assignment focuses on using classes,inheritance concept and virtual function's(if needed).

Some sections have possible deductions that would exceed the points allocated for that section. Mark and comment each possible deduction but, when totaling them, deduct no more than the total points allocated for that section. Record the awarded points on the accompanying grading sheet.

**Program Style** **25 points**

Sufficient Program and Function Header Comments **5 points**

- Deduct **2 points** each for not including student's name, TA's name
- Deduct **2 points** each for not including course, date, garnet ID, program number
- Deduct **2 points** each for omitting summary, input, output descriptions, data structures or assumptions.
- Deduct **1 point** for each section that is shortened to the point that it describes the program inadequately.

*Deductions should total no more than 5 points*.

Internal comments describe flow of program **5 points**

Internal comments should label the major sections of the program *and* explain line-level details e.g. description of variables and constants. Specifically, make sure that the C++ Style Guidelines for COP 3330 (available in the handouts section of the course homepage) have been followed. **NOTE**: Both the main function AND any other functions must be commented.

- Point out deficient internal comments; deduct **3 points** if you see a pattern
- Deduct **5 points** if *all* internal comments are missing or seriously lacking
- Deduct **2 points** per incident where critical or important internal comments missing or insufficient (e.g. where a function call is made)

*Deductions should total no more than 5 points*

Vertical and horizontal spacing enhance readability **5 points**

- Deduct **2 points** for consistently failing to indent statements
- Deduct **2 points** for not using blank lines to separate sections of the program or comments
- Deduct **2 points** for not aligning internal comments (also deduct if comments missing)
- Deduct **2 points** if the maximum number of columns exceeds 80
- Deduct **2 points** for not give the semantic meaning of function's arguments and return value.

*Deductions should total no more than 5 points*

Well-named Identifiers and Correct use of constants to facilitate maintainability **5 points**

- Deduct **3 point** each for not using constants .
- Deduct **2 points** for each poorly named identifier e.g. x, m1, etc.
- Deduct **2 points** for declaring a constant and not using it.

*Deductions should total no more than 5 points*

Output well-formatted and easy to read **5 points**

- Deduct **3 points** for missing or inappropriate opening or closing message
- Deduct **2 points** for cutesy/ rude/ inappropriate messages
- Deduct **5 points** for overly long output
- Deduct **5 points** if the output is not properly spaced (horizontally and vertically)
- Deduct **5 points** for not giving out proper error or help information if the user input is invalid.

*Deductions should total no more than 5 points*

**Program coding and output** **50 points**

Program is complete and compiles (no syntax errors) **25 points**

- Deduct **20 points** if the program compiles, but is not complete
- Deduct **10 points** if the program compiles but does not run (link errors)
- Deduct **10 points** if program crashes or becomes frozen or dead looped at any stage
- Deduct **2 points** for each relevant syntax warning (up to 10 points)

*Deductions should total no more than 25 points*

Program Input **30 points**

- Deduct **10 points** if program does not check for the file.
- Deduct **10 points** if they don't end the battle after 20 rounds.

*Deductions should total no more than 10 points*

Program produces correct output 3**0 points**

- Deduct **5 points** if not echo printing contents of the file.
- Deduct **10 points** if the output is incomplete.

*Deductions should total no more than 30 points*.

Program generally well-coded **10 points**

- Deduct **2 points** each for unreachable code e.g. if/ else always false, function that is never called, etc.
- Deduct **2 points** each for unneeded loops, condition checking, etc.
- Deduct **2 points** for logical errors e.g. adding instead of subtracting, comparing the wrong variables, using && instead of ||, etc.
- Deduct **2 points** for generally unorganized code.

*Deductions should total no more than 10 points*

**Program efficiency and structure** **20 points**

Correct use of functions to perform useful tasks **20 points**

- Deduct **up to 30 points** if inheritance concept is not used.
- Deduct **2 points** each for ignoring the return value of user-defined value-returning function
- Deduct **2 points** for not having a function for sections of code that are repeated

*Deductions should total no more than 10 points*

Correct use of selection, iteration and branching **10 points**

- Deduct **5 points** if program does not loop properly
- Deduct **2 points** each for a loop that does not follow the style guidelines. The best loop for the task, as defined by the student's code should be chosen. Refer to the style guide for information on poor looping constructs.
- Deduct **2 points** for each occurrence of extra {}'s except when extra {}'s clarify the structure of nested-if's
- Deduct **2 points** each for excessive use of return statements
- Deduct **2 points** each for poor use of break and continue
- Deduct 5 points for not using width or height properly within the looping of setting value of each pixel.
- Deduct **10 points** for ANY use of goto

*Deductions should total no more than 10 points*

**Other Penalties**

Late penalty **20 points**

- ***LATE PROGRAMS SUBMIT NOT MORE THAN 2 DAYS, DEDUCT 20 POINTS!***

Does not compile **50 points**

- Deduct **50 points** if program does not compile

Misnamed file or multiple files submitted **10 points**

- Deduct **2 points** if file name is capitalized or plural (i.e. CreatureApp.cpp, CREATUREAPP.cpp, …)
- Deduct **5 points** if tar file creature.tar is not tarred properly.

*Deductions should total no more than 10 points*

Commands

**40 points**

- Deduct up to **10 points** if attack and defend are not done as specified in the writeup.
- Deduct up to **10 points** if they don't terminate the game if they run out of creatures.
- Deduct up to **10 points** if they dead guy attack again.
- Deduct up to **5 points** for closing the program abruptly.
- Deduct up to **5 points** if they create an instance of the demon class.
- Deduct up to **5 points** if they don't set the hit point to 0 if the creature is dead.

**Bonus Points** **maximum of 10 points**

Make sure that the student has implemented all basic requirements as specified in the writeup and it is up to the grader to decide the extra bonus point **if needed!!!**

**15 points**

- Deduct **5 points** each for any global variables are used (global constants are fine)

*Deductions should total no more than 15 points*

Correct use of class and data structures **40 points**

- Deduct up to **15 points** if they violates some principles of the COP 3330 course.
- *Deductions should total no more than 50 points*

No Functions **35 points**

- Deduct up to **25 points** for not using sufficient functions. This would be indicated by an overly long main or 1 or more very long functions.
- Deduct up to **20 points** for not using any functions that pass parameters or for not using any functions that return values.

*Deductions should total no more than 35 points.*

Automatic zeros **100 points**

- Deduct **100 points** if program contains another student's name or plagiarized code .

These issues raise suspicion of violation of the FSU Academic Standards and Policies. Please report such incidences immediately.

**DO NOT GRADE SUBMISSIONS THAT ARE NOT C++ SOURCE CODE!**