

Trent University
Computer Science 204H
2007 Fall Term

OBJECT-ORIENTED PROGRAMMING

Assignment Grading Guidelines

To avoid confusion, here is the list of requirements for COSC 204 assignments:

- Each assignment will be marked out of 10
- Assignments will be considered separately (so if you do the same thing incorrectly in each assignment, you will be penalized each time)
- The smallest deduction will be 1/2 mark
- The following components will be considered (and their respective weights)

Correctness 50% - this corresponds to how well your program works (i.e. creates the correct output). Correctness covers all parts of the question:

- Code to produce output
- Code to test for input errors
- Parameters and parameter passing (if applicable)
- You will be marked not only on correctness, but also on efficiency.
- Making good choices when it comes to programming constructs like for example using ELSE IF's instead of multiple IF's

Documentation (15%) - part of being a good programmer is explaining your program to others:

- A program description at the top of the program
- A data dictionary (a listing of the variables and what they represent)
- A short description of each subroutine at the beginning of each subroutine (if applicable)
- A description of each parameter used by the subroutine (if applicable)
- Inline comments (i.e., you should have comments spread throughout your program explaining what is happening)
- External documentation (if applicable) includes pseudo code or flowchart, and a structure chart

Style (10%) - use of descriptive identifier names (variables, subroutines, etc.):

- Index and temp variables can be cryptic but main identifiers must be self-documenting
- Proper use of indentation to improve the readability of your program, i.e. indented loops, selection and compound statements
- Use of blank lines to improve readability

Testing (15%) - must prove that your program works:

- One test is not enough!
- Test all input cases (including invalid input)
- You can assume that if the program is looking for a character, you will supply it a character (it may just not be the correct one)
- Be sure to test the end cases
- In addition to the actual testing, include separate Test documentation that consists of the input choices, the expected output, and the purpose of the test (can actually write this on the output itself if you wish)

Output (10%) - presentation of your output

- Showing an appropriate number of decimal places
- Indicating currency if necessary
- Lining up columns in a table

Notes:

- Marks will be deducted from each category until it reaches 0 (e.g., since each deduction is worth 1/2 mark, then after 3 omissions in your documentation component you get 0)
- Marks will not be deducted for the same mistake within a question (e.g., if you did not indent at all, you will only have one deduction)
- It is possible to have a perfectly working program and not receive a good mark
- Any program that works correctly will receive at least 50%
- All attempts (even if they don't work) will be assigned part marks
- These are your marks and you are entitled to a re-mark if you are not happy.