2024-09-19

# Rubric

Below are the rubrics used to grade each phase of the “Character creation” project. It is advised to use these rubrics as a way to check your work, along with the solution for the corresponding phase.

## Phase 1

| Criteria | Points |
| --- | --- |
| **In**correct filename (lname\_fname.zip, not lname\_fname.zip.zip for instance) | -0.5pt |
| **In**correct file (archive zipped, not just the .cs file) | -0.5pt (if the student did not share their .cs code at all, then your grading ends here) |
| Name and date in a delimited comment **not** at the very beginning of their code | -1pt (If they forgot the date, their name, if the comment is in-line, that is // and not /\* ... \*/, or if it is not the very first thing in the file, they get only -0.5pt. If they made multiple mistakes (eg, only their name, in the main method), they get -1pt) |
| If the program does not compile | Stop here. No points. |
| Correctly declared the attributes | +1pt |
| Correctly implemeted the getters | +1pt |
| Correctly implemented the setters (Aside from SetHitPts) | +1pt |
| Correctly implemetented the SetHitPts and LevelUp | +1pt (0.5pt per method) |
| Correctly implemented a ToString | +1pt |
| Correctly implemetented the empty and populated Constructors | +1pt (0.5pt per method) |
| **Total (with Bonus)** | **6** |

### Additional Info

* For all the getters, setters, and attributes, every error is -0.5pt, with more than one error meaning 0pts in that section. (Do NOT give quarter points)
* For every attribute with an improper name or datatype, take 0.5pt off the 1pt (that is, 2 mistakes result in a 0)
* If the getter or setters are correct w.r.t. the name / datatype of the attribute as implemented by the student, please do not take points off.
* *Any* ToString is considered valid, as long as it compiles and returns a string.

## Phase 2

| Criteria | Points |
| --- | --- |
| **In**correct filename (lname\_fname.zip, not lname\_fname.zip.zip for instance) | -0.5pt |
| **In**correct file (archive zipped, not just the .cs file) | -0.5pt (if the student did not share their .cs code at all, then your grading ends here) |
| Name and date in a delimited comment **not** at the very beginning of their code | -1pt (If they forgot the date, their name, if the comment is in-line, that is // and not /\* ... \*/, or if it is not the very first thing in the file, they get only -0.5pt. If they made multiple mistakes (eg, only their name, in the main method), they get -1pt) |
| If the program does not compile | Stop here. No points. |
| Correct empty constructor | +1pt (-0.5pt per error) |
| Correct populated constructor | +1pt (-0.5pt per error) |
| Collects name | +0.5pt |
| Collects race (must be Human, Elf, Orc, Dragonborn, Tiefling) | +1pt (-0.5pt if case sensitive, -0.5pt if mostly correct condition, -1pt otherwise (including no validation), -0.5pt per any other error) |
| Collects gender (must be (M)ale, (F)emale, (O)ther, (H)ermaphroditic) | +1.5pt (-0.5pt if case sensitive, -0.5pt if mostly correct condition, -1pt otherwise (including no validation), -0.5pt per any other error) |
| Collects level (between 0 and 30, inclusive) | +1pt (-0.5pt if exclusive, -0.5pt if mostly correct condition, -1pt otherwise (including no validation), -0.5pt per any other error) |
| Bonus: Tieflings can only be hermaphroditic | +0.5pt (no points if anything wrong) |
| Bonus: Always store gender as uppercase | +0.5pt (no points if anything wrong) |
| If Character is not created after collecting values | -0.5pt |
| **Total (with Bonus)** | **7** |

## Phase 3

| Criteria | Points |
| --- | --- |
| **In**correct filename (lname\_fname.zip, not lname\_fname.zip.zip for instance) | -0.5pt |
| **In**correct file (archive zipped, not just the .cs file) | -0.5pt (if the student did not share their .cs code at all, then your grading ends here) |
| Name and date in a delimited comment **not** at the very beginning of their code | -0.5pt (If they forgot the date, their name, if the comment is in-line, that is // and not /\* ... \*/, or if it is not the very first thing in the file, they get only -0.5pt) |
| If the program does not compile | Stop here. No points. |
| Creates stats array and role attributes | +0.5pt (No deduction if no array is used, any mistake is -0.5pt for this section |
| Modified ToString to include charclass (and job for bonus) | +0.5pt (no deductions if job is missing, -0.5pt if class is missing) |
| AssignStats method | +2pts (see below for in-depth deductions) |
| SetRole method | +2pts (see below for in-depth deductions) |
| Dice rolls are generated and assigned to int array | +1pt (-0.5pt if array is not used, -0.5pt if random is not used or if used incorrectly, -0.5pt if bounds on random is incorrect) |
| SetJob method | +2pts (see below for in-depth deductions) |
| Modify dice rolls to be 4d6 best 3 of 4 | +1pt |
| **Total (with Bonus)** | **9** (Max Points is 8, even if all bonus points are completed) |

### Additional Info

* AssignStats
	+ -0.5pt per error in decision structure
	+ -0.5pt if no input validation
	+ -0.5pt if you can overwrite previous stat assignments
* SetRole
	+ -0.5pt if no array is used to store possible classes
	+ -0.5pt per error in conditions
	+ -0.5pt if able to select class I shouldn’t be able to
	+ -0.5pt if no validation
	+ -1pt if multiclasses are significantly broken
* SetJob
	+ -0.5pt if no array is used to store possible jobs
	+ -0.5pt per error in conditions
	+ -0.5pt if able to select class I shouldn’t be able to
	+ -0.5pt if no validation