1. clean main (i.e. nothing in the main that shoud be in a class)
2. useful class (explain why)
3. useful and correct encapsulation (explain why)
4. useful and correct inheritance (explain why)
5. useful and correct object composition (explain why)
6. useful and correct multiple inheritance (explain why)
7. correct base class
8. correct abstract base class
9. useful virtual function
10. useful proven (dynamic) polymorphism
11. at least 2 default constructors
12. at least 2 specific constructors
13. member initialization in constructors (the stuff behind a colon)
14. at least 2 destructors
15. useful useage of "this" (if the code does not work without it)
16. useful member function
17. default values in function definition
18. useful member variabel
19. useful getters and setters for member variables
20. correct protections
21. no mistake in object-oriented programming
22. no globals, but statics if needed
23. everything in one or more self-made namespace(s)
24. maintanability by clean uniform code style and good function naming and/or comments everywhere
25. seperate header files
26. 2 useful unsigned chars or other better usage of memory efficient type
27. using the best suited type on a lot of places (almost everywhere possible)
28. at least 4 useful const references
29. correctly using const in almost all places (almost everywhere possible)
30. at least 4 useful bool
31. useful pointer declaration and usage
32. dynamic memory allocation (new)
33. dynamic memory removing (delete)
34. 2 useful (modern) call-by-references
35. useful string class usage
36. useful container class
37. useful Qt class
38. useful operator overloading
39. useful function overloading (non operator)
40. useful function overriding (non virtual)
41. useful template function or class
42. useful non-type template arguments
43. useful friend function or class
44. correct useage of inline function
45. useful recursive function
46. useful usage of nullptr
47. useful usage of (modern) file-I/O
48. correct and useful usage of command line parameters
49. useful usage of signals/slots
50. useful usage of lambda function
51. one complete project that compiles and does not crash
52. working build manual as readme on GitHub (project must be possible to build from scratch on a clean PC)
53. test-driven development (= written test plan or unit tests)
54. solved on 8 bug tickets
55. reported 8 bug tickets on other projects
56. usage of a GUI
57. usage of OpenGL or other 3D engine
58. useful usage of an external library (not Qt)