

1. name at least 5 variable types in c++
2. what is "int overflow"?
3. why ``using namespace std`` might not be a good idea?
4. which header do you have to include for the access to `std::cout`?
5. what would be the simplest "legal" program in c++?
6. what is the difference in c++ between ``struct`` and ``class``?
7. what does ``break`` statement do in a loop in c++?
8. what does ``continue`` statement do in a loop?
9. how do you define a static array of type `int` and size 7?
10. what can you use to read interactive user input?
11. What is a reference?
12. What is a difference between passing variables to functions by reference and by value?
13. Why do we want to pass variables to functions by reference rather than by pointer?
14. If the function accepts pointers (signature ``int function(int* a)``), how do you pass a variable defined as ``int x`` to it?
15. Why do we want to use ``const`` modifier whenever possible?
16. If you have a pointer named ``p``, how do you access a value it points to? What is the name of that process?
17. What value is stored in the pointer variable itself?

18. What can you say about a function that has signature `void function()`?
19. What is an `auto` keyword?
20. What is function overloading?
21. What is a recursive function?
22. What is the role of a constructor in a class?
23. How can we change private member variables of a class?
24. How do you create an object file with `g++`?
25. Why can't you just have everything in one file?
26. What is the purpose of a "header guard"?
27. What does `-IFOLDER_NAME` mean when passed to `g++`?
28. Do tabs matter in a `Makefile`?
29. What does `$$` mean in a `Makefile`?
30. What does `^` mean in a `Makefile`?
31. How to use `make` with a makefile that is not named `Makefile`?
32. What does it mean if a function is a friend of a class?
33. Why should non-class member operators be friends of classes?
34. Which header do you need to include to work with files?
35. How do you open a file in append mode?

- 36. how do you allocate a dynamic array in C++?
- 37. whats the difference between ``delete`` and ``delete[]``?
- 38. when do you need to overload assignment operator for your class?
- 39. when do you need to create a copy constructor for your class?
- 40. when do you need to create a move constructor for your class?
- 41. what should you do if your class allocates resources, but you are sure you will never need a copy constructor?
- 42. what are protected class members?
- 43. what are virtual functions?
- 44. what is an abstract class?
- 45. explain dynamic (runtime) polymorphism
- 46. why destructors should be made virtual?
- 47. what is this ``[](){}()```?
- 48. what's the general structure of a lambda function?
- 49. what does "mutable" keyword do in a lambda function?

- 1. How do you create markdown cell in Jupyter? (with keyboard commands)
- 2. How do you delete a cell in Jupyter (with keyboard command)?
- 3. How do you add a cell above the current one in Jupyter (with keyboard command)?

4. What advantage does conda environment provide over a system-wide installation of python?
5. What does it mean that variables are dynamically typed in python?
6. What's the difference between python and C/C++ integers?
7. Explain the difference between lists, sets and tuples in python.
8. What does negative index mean when accessing list elements?
9. How do you print the first N elements of a list?
10. How do you print every Nth element of a list?
11. How do you print the reverse list?
12. Give an example of a list comprehension.
13. Give an example of cell and line magic commands in Jupyter.
14. Write a "hello world" function in python (syntax errors are not ok in hello world even on paper).
15. What are keyword function arguments in python and what is their advantage?
16. Why shouldn't you do ``from numpy import *``?
17. What is the advantage of using numpy arrays over lists? What are the dangers?
18. What's the most widely used package for plotting in python?
19. What's the purpose of ``__str__`` method in a python class?

20. What package can you use in python for symbolic calculations?
21. What are decorators? Write an example (minor syntax errors are ok here)
22. Name two ways of calling c++ code from python.
23. What is pandas? Name at least 5 functions from that package.