

OOP for Scientific Computing First Exam

points:

1	2	3	4	5	Σ
15	20	10	10	10	65

Following was asked in the exam:

1. Explain following aspects of C++ :
 - a. `class` vs `struct`?
 - b. `namespace`. What is it, when & why is it used?
 - c. `private` vs `protected`?
 - d. copy elision?
 - e. header guards. What, why?
 - f. rule of five vs rule of zero?
 - g. temporaries? literals?
 - h. how does a shared pointer work?
 - i. SFINAE?
 - j. SOLID?
2. Short code snippets were given and asked to explain & extend. With following topics:
 - a. default function args
 - b. concepts
 - c. template parameters, default template parameters
 - d. Constructor, copy constructor, copy assignment operator, move constructor, move assignment operator, overloaded constructor.
 - e. Lambda expression, functional programming
 - f. Compile time branching
 - g. CRTP
 - h. Inheritance. How to improve the given implementation
3. Horner Schema: Variadic templates, recursion with templates(?), template metaprogramming
4. An incomplete implementation of an OOP system was given that was supposed to implement a simulation a prey-predator dynamic system, modeled by the Lotka-Volterra differential equations:

$$\frac{dx}{dt} = \alpha x - \beta xy,$$

$$\frac{dy}{dt} = \gamma y - \delta xy$$

The incomplete OOP system had the following basic structure:



the method "Simulate()" returns an object of type "PreyPredatorData", and is supposed to populate its members with the simulation data based on discrete Lotka-Volterra equations.

in a) we were supposed to write an implementation for this method.

Figure 1: Lotka-Volterra OOP System

- a) Implement the `PreyPredatorData Simulate(int steps, double dt)` that creates a `PreyPredatorData` object, populates its members with the simulation data based on the Lotka-Volterra difference equations and returns the object:

$$X_{n+1} = X_n + \Delta t(\alpha x_n - \beta x_n y_n)$$

$$Y_{n+1} = Y_n + \Delta t(\delta x_n y_n - \gamma y_n)$$

b) ?

5. ?