

[1*] 1. Consider the following two implementations of the Point class:

```
// implementation #1
class Point { // 2D point concept
public: // Constructor(s)
    Point(int xx, int yy) { x = xx; y = yy; }
    Point() { x = y = 0; }
    // ...
private: // Implementation
    int x, y;
};

// implementation #2
class Point { // 2D point concept
public: // Constructor(s)
    Point(int xx = 0, int yy = 0) { x = xx; y = yy; }
    // ...
private: // Implementation
    int x, y;
};
```

Which one do you prefer? Please note, both implementations are totally correct from the point of C++ syntax/semantics view.

Hint: focus on constructing objects using each constructor.

[2*] 2. Define and implement classes for representing *time* and *rational numbers*. declare and define constructors and some simple member functions as interfaces. Use default arguments for reducing the number of constructors.

[2*] 3. In session 6 we implemented stack as a component using struct. Convert it to a class. Declare and define constructors for it.

Hint: use the stack0 project, we sent it to the boot camp group.

[1*] 4. Consider the following class:

```
class X {
public:
    int mf(int v);
private: // Implementation
    int m;
};
```

Write the body of the mf member function, so set m with v, but return the old value of m.

[1*] 5. Represent the type Student_info abstraction to represent the information of a typical student. Each student has a name and three grades: midterm, final, and some unknown number of home works. The grades are double-precision floating points. Hint: we need just the implementation of such abstraction.