

Advanced Programming Techniques

Tasks of a model exam

Task 1: Definitions and Terms

- (b) (4P) Explain the difference between static and dynamic polymorphism and state by which two concepts they are realized in C++.

- (c) (2P) Given is the function `fct` which takes a pointer to a `double` and a `const` reference to a `std::list` as input and returns no value.
Using the `std::function` library type, define the variable `f` and initialize it with `fct`.

Task 2: Programming with the Standard Library

Please note: The questions assume that all necessary header files from the Standard Library are included and an implicit `using namespace std;`. Likewise, you can safely assume the same for your code!

Your task is to implement the function

```
template<typename T>
list<T> eraseDuplicates(const list<T> & input)
```

- this function should return a list consisting only of unique elements of the passed list `input`
- preserving the order of the `input` list is not important
- you can assume that instances of `T` can be compared for equality

Example usage:

```
list<string> l = {"a", "c", "b", "a",
                "b", "b", "a",
                "c", "c"};
```

```
auto res = eraseDuplicates(l);
```

```
res.sort();
for(const auto & e : res)
    cout << e << endl;
```

Expected output of above snippet:

```
a
b
c
```

Task 3: Vector Class

Consider the following implementation of a `Vector` class:

```
class Vector {
private:
    int size_;
    double * data_;

public:
    Vector(int size) : size_(size), data_(new double[size]) {}

    ~Vector() { delete [] data_ }

    Vector& operator=(const Vector & o ) {
        this->data_ = o.data_;
        this->size_ = o.size_;
    }
}
```

- (a) (1P) List the three aspects of C++'s "Rule of Three".
- (b) (2P) Explain why is the current assignment operator wrong in terms of the Rule of Three in the current (unmodified) `Vector` class.
- (c) (6P) Fix and extend the `Vector` class, to make it comply to the "Rule of Three". All added functionality has to work correctly for vectors with different sizes.
- (d) (6P) Extend the `Vector` class such that the following code compiles and correctly prints the vector elements 0 and 1 to `stdout`.

```
void printVector(const Vector & v) {
    for(auto it = v.begin(); it != v.end(); ++it)
        std::cout << *it << std::endl;
}

int main() {
    Vector v(2);
    v(0) = 0;
    v(1) = 1;

    printVector(v);

    return 0;
}
```