Dr. Harald Köstler
Jan Hönig

Winter Term
2017/2018

Exam in

Advanced Programming Techniques

March 15, 2018

Name: _____

Date of birth: _____

Student number: _____

Would you like your grade to be published on the chair's notice board together with the last five digits of your student number?

☐ Yes / ☐ No

Available time: 60 min
Available points: 100

Please do not fill out anything below this line!

Total number of points: _____ of 100

Grade: _____

Passed: ☐ Yes / ☐ No

Problem 1: Definitions and Terms (13 points)

(a) (4P) List the following:

Two built-in types: _____

_____

Two associative containers: _____

_____

Two C++11 or 14 features: _____

_____

Two C++ keywords for type casts: _____

_____

(b) (3P) 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`.

_____

_____

_____

_____

(c) (6P) List three different parts of the standard library (e.g. by specifying the corresponding header file) and a concrete class defined within.

_____

_____

_____

_____

_____

_____

_____

Problem 2: Errors and Function Matching (14 points)

Given is the following **incorrect** C++ program that should output *Bar int doit()*:

```
#include <iostream>

struct Foo {

    Foo(const Foo&) {}

    virtual int doit() { std::cout << "Foo int doit()" << std::endl; return 1; }

    virtual char doit() { std::cout << "Foo char doit()" << std::endl; return 1; }

};

class Bar : Foo {

    int doit() { std::cout << "Bar int doit()" << std::endl; }

};

void main() {

    Bar f;

    Foo& fooref = &f;

    fooref.doit();

}
```

List all six errors in the code and state how one can change the program such that the correct output is printed. (**Hint**: Check for accessibilty, function matching and inheritance)

Problem 3: Programming with the Standard Library (7 points)

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 new 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 sorted

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
```

Problem 4: Vector Class (20 points)

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) (2P) Explain why the current assignment operator is wrong in terms of the Rule of Three in the current (unmodified) `Vector` class.

(b) (7P) Implement a correct version of the assignment operator and a copy constructor for the `Vector` class. All added functionality has to work correctly for vectors with different sizes.

(c) (5P) Extend the `Vector` class such that the following code compiles and correctly prints the vector elements 0 and 1 to `stdout`.

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

  const Vector v2(v);

  std::cout << v2(0) << std::endl;
  std::cout << v2(1) << std::endl;

  return 0;
}
```

(d) (6p) Extend the `Vector` class such that the following code compiles and correctly assigns and prints the vector elements to `stdout`.

```
int main() {
  int i = 0;
  Vector v(42);
  for(double* it = v.begin(); it != v.end(); ++it)
      *it = i++;

  const vector v2(v);

  for(const double* it = v2.begin(); it != v2.end(); ++it)
      std::cout << *it << std::endl;

  return 0;
}
```

Problem 5: Questions on the Project (6 points)

You are required to reference your project code in this task (*not* the assignment code) where indicated. Please provide line numbers as well as the file you are referring to.
Give two examples where you made use of C++11, C++14 or C++17 features in your project code. If you didn't use any such features give two examples where they *could* be used.

Discuss how this improved (would improve) code quality, e.g. by allowing a more concise formulation.

Problem 6: Project (40 points)

Please make sure that you submit your project code with the exam paper such that we are able to check it! In case two groups submit the same code none of them will obtain any points for it.

(a) Which race did you implement in your group?

(b) What is your group's name?

(c) Did you pass the forward simulation task?

☐ Yes / ☐ No

(d) Did you pass the optimization task?

☐ Yes / ☐ No

(e) Did you pass the push challenge?

☐ Yes / ☐ No

(f) Did you pass the rush challenge?

☐ Yes / ☐ No