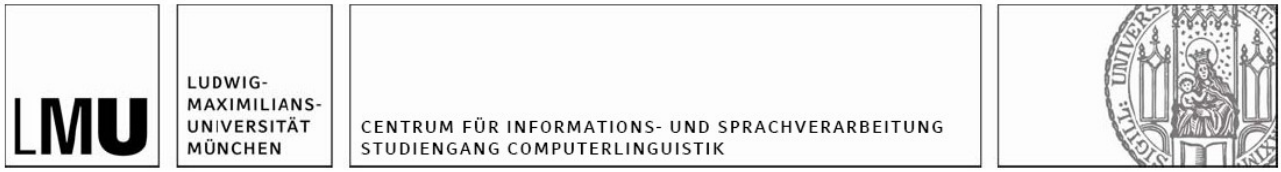

Klausur zur Vorlesung „Höhere Programmierung" im SS 2011

Andreas Neumann M.A.

Datum : 25.07.2011

Vorname:	
Nachname:	
Matrikelnummer:	
Studiengang:	

Die Klausur besteht aus **6 Aufgaben**. Die Punktzahl ist bei jeder Aufgabe angegeben. Die Bearbeitungsdauer beträgt **60 Minuten**. Bitte überprüfen Sie, ob Sie ein vollständiges Exemplar erhalten haben.

Tragen Sie die Lösungen in den dafür vorgesehenen Raum im Anschluss an jede Aufgabe ein.

Bewertung

Aufgabe	Erreichte Punktzahl	Mögliche Punktzahl
Aufgabe 1 – IO		11
Aufgabe 2 – Frequenzliste		16
Aufgabe 3 – Objektorientierte Programmierung		20
Aufgabe 4 – Collections, STL		21
Aufgabe 5 – Boost		18
Aufgabe 6 – Syntaxfehler finden		14

Gesamtpunktzahl		100
Note		

Einwilligungserklärung

Hiermit stimme ich einer Veröffentlichung meines Klausurergebnisses unter Verwendung meiner Matrikelnummer im Internet zu.

_____ (Name, Datum)

Aufgabe 1 – IO

Was geben folgende Programmfragmente auf dem Terminal aus? Sie können davon ausgehen, dass das Terminal UTF-8-fähig ist und alle locales korrekt gesetzt wurden.

Teilaufgabe 1.1 - Ausgabe mit ISO-8859-X

Programmfragment	Erwartete Ausgabe auf dem Terminal
`char wort[] = {'A','B','C'};` `cout << wort[2] << endl;`	
`double zahl = 42.7;` `cout << zahl << "," << (int) zahl << endl;`	

`string einWort = "Eis";`
`string nochEinWort = "verkaeufer";`
`cout << einWort << nochEinWort << endl;`
`cout << (einWort + nochEinWort).at(3) << endl;`	

`/* Großbuchsaben im ASCII-Bereich beginnen bei 65`
`und Enden bei 90 */`
`string name = "HAL";`
`int i;`
`char c;`
`for ( i = 0; i < 3 ; i++ ) {`
`  c = name.at(i);`
`  cout << ++c;`
`} ;`
`cout << endl;`	

Teilaufgabe 1.2 - Ausgabe mit Unicode

Programmfragment	Erwartete Ausgabe auf dem Terminal
`wstring ws = L"ˈjuːnɪkoʊd";` `wcout << ws[4] << endl;`	

`wstring xElementN( L"x∈ℕ" );`
`wcout << xElementN.length() << endl;`
`wcout << xElementN.at(2) << endl;`	

`/* Unicode-Alpha  'A'  ->  913`
`  ASCII  'A'  ->  65`
`*/`
`wchar_t alphaOrA = L'A';`
`wcout << alphaOrA << endl;`
`wcout << (int) alphaOrA << endl;`	

Punkte (__ / 11)

Aufgabe 2 – Frequenzliste

Teilaufgabe 2.1 – Erstellen einer Frequenzliste

Schreiben sie ein komplett lauffähiges C++ - Programm, welches die Frequenzliste eines in utf-8-kodierten Textes namens **einText.txt** erzeugt.

Teilaufgabe 2.2 – Ausgabe einer Frequenzliste

Erweitern sie das Programm dahingehend, dass nach Erzeugung der Frequenzliste diese durchlaufen und auf einem utf-8-fähigen Terminal ausgegeben wird.

Punkte (__ / 16)

Aufgabe 3 – Objektorientierte Programmierung

Schreiben sie eine Klasse **Film** mit den privaten Attributen **wstring titel** , **double spieldauer** und **bool gesehen**. **titel** und **spieldauer** sollen als Attribute dem Konstruktor der Klasse übergeben werden.
Eine öffentliche Methode **anschauen** soll den Wert von **gesehen** von **false** auf **true** setzen.
Denken sie an die „include guards“.

Teilaufgabe 3.1 Header – Film.h

Teilaufgabe 3.2 Implementierungsdatei – Film.cpp

Teilaufgabe 3.3 Main-Methode – main.cpp

Erzeugen sie zwei Instanzen des Objekts Film. Rufen sie die Methode **anschauen** bei der ersten Instanz des Objekts auf.

Teilaufgabe 3.4 Kompilieren des Programms

Geben sie den Befehl zum kompilieren des Programms aus Aufgabe 3 auf der Konsole an.

Punkte (__ / 20)

Aufgabe 4 – Collections, STL

Teilaufgabe 4.1

Schreiben sie ein komplett lauffähiges Programm welches die Zahlen von 1 - 100 000 in ein Array und in einen Vektor einliest.

Teilaufgabe 4.2

Durchlaufen sie den Vektor mit einem Iterator und geben alle Zahlen aus, die durch 42 ohne Rest teilbar sind.

Teilaufgabe 4.3

Durchlaufen sie das Array rückwärts und bilden sie die Summe aller Zahlen im Array. Geben sie bei jedem tausendsten Element ein Zwischenergebnis aus.

Punkte (__ / 21)

Aufgabe 5 – Boost

Füllen sie die Leerstellen mit den korrekten Codefragmenten um ein lauffähiges Programm zu erzeugen.

Teilaufgabe 5.1

Gegeben ist folgendes Programm, welches einen Text an Leer- und Satzzeichen aufsplittet und in einem späteren Schritt die einzelnen Tokens auf dem Terminal ausgibt.

```
#include <string>
#include <vector>
#include <iostream>
_________________
_________________

using namespace std;
_________________

int main() {
	setlocale(LC_ALL,"");
	_________________ yoda = L"Yes. A Jedi's strength flows from the Force. But
beware the dark side. Anger, fear, aggression. The dark side of the Force are
they. Easily they flow, quick to join you in a fight!";
	vector<_________________> words;
	split( _________________, yoda, _________________, token_compress_on );

	BOOST_FOREACH( wstring word, _________________) {
		wcout << _________________ << endl;
	}}
```

Teilaufgabe 5.2

Gegeben ist folgendes Programm, welches anhand von regulären Ausdrücken bestimmt, ob ein Namen einem bestimmten Kriterium entspricht und dazu einen passenden Satz ausgibt.

```
#include <iostream>
#include <string>
#include <boost/foreach.hpp>
#include _________________
_________________
int main() {
	string _________________ = {"Luke","Leia","Han","Chewbacca"};
	boost::regex vierBuchstaben(_________________);
	boost::regex beginntMitC(_________________);
	BOOST_FOREACH( _________________, namen) {
		if ( boost::regex_match( name, _________________ ) ) {
		   cout << name <<"! The force is strong in this one. " << endl;
		}
		if ( _________________ ( name ,beginntMitC ) ) {
		   _________________ << name <<  " is a Wookiee!" << endl;
		}}}
```

Punkte (__ / 18)

Aufgabe 6 – Syntaxfehler finden

Unterstreichen und korrigieren sie die vorhandenen Syntaxfehler.

Teilaufgabe 6.1 (8 Syntaxfehler)

```
#include<iostream>
include<list>
#include<algorithm>
using std namespace;

void putOut(string);
list[string] buildList();

int main() {
   list<string> myList = buildList()

   for_each( myList.begin() ; myList.end(), putOut() );
}

void putOut(string ding) {
   cout >> ding >> endl;
}

list<string> buildList() {
   list<string> myList = list<wstring>();
   myList.push_back("1");
   myList.push_back("2");
   myList.push_back("drei");
   myList.push_front("null");

   return myList;
}
```

Teilaufgabe 6.2 (6 Syntaxfehler)

```
#include <string>
#include <iostream>
#include "boost/foreach.hpp>

using namespace std;

int main[] {
   setlocale(LC_ALL, "');
   wstring unicode( "['ju:nɪkoʊd]" );

   BOOST_FOREACH( wchar_t ch  unicode ){
       wcout << ch  << endl;

}
```

Punkte (__ / 14)

Zusatzblatt

Nutzen Sie dieses Blatt, wenn ihnen der Platz ausgeht.