

Matrikelnummer:

Kursbezeichnung: Höhere Programmierung in der Computerlinguistik mit C++

Dozent: Andreas Neumann

1. Was geben folgende Programmfragmente bei ihrer Ausführung auf dem Terminal aus?

|    | Quelltext   | Ausgabe                                |
|----|---|--|
| a) | <pre>int var1 = 5; cout &lt;&lt; var1 &lt;&lt; endl;</pre>  | 5                                      |
| b) | <pre>double var2 = 4.9; cout &lt;&lt; var2 &lt;&lt; endl; cout &lt;&lt; (int) var2 &lt;&lt; endl;</pre>   | 4.9<br>4                               |
| c) | <pre>string var3 = "Baumschule"; cout &lt;&lt; var3 &lt;&lt; endl; cout &lt;&lt; var3.at(1) &lt;&lt; endl;</pre>  | Baumschule<br>a                        |
| d) | <pre>wstring var4 = L"Bürger"; wcout &lt;&lt; var4 &lt;&lt; endl; wcout &lt;&lt; var4.at(1) &lt;&lt; endl;</pre>  | Bürger<br>ü                            |
| e) | <pre>/*Unicode Codepoints:  * α -&gt; U+03B1 = 945  * β -&gt; U+03B2 = 946  * γ -&gt; U+03B3 = 947  */ string var5      = "αβγ"; wstring var5_2   = L"αβγ"; cout &lt;&lt; var5 &lt;&lt; endl; wcout &lt;&lt; var5_2 &lt;&lt; endl; wcout &lt;&lt; var5_2.at(1) &lt;&lt; endl; wcout &lt;&lt; (int) var5_2.at(1) &lt;&lt; endl; cout &lt;&lt; var5.length() &lt;&lt; endl; wcout &lt;&lt; var5_2.length() &lt;&lt; endl;</pre> | αβγ<br>αβγ<br>β<br>946<br>> 3 → 6<br>3 |
| f) | <pre>char var6 = 'a'; //ASCII 97 cout &lt;&lt; var6 &lt;&lt; endl; cout &lt;&lt; (int) var6 &lt;&lt; endl; wcout &lt;&lt; var6 &lt;&lt; endl;</pre>   | a<br>97<br>a                           |

( / 16)

2. Lesen Sie die utf-8-kodierte Datei "namen.txt" und speichern sie deren Inhalt in einem Container der STL.

```
#include <fstream>
#include <string>
#include <vector>
using namespace std;

int main() {
    setlocale(LC_ALL, "");
    locale unicode("de_DE.UTF-8");
    wfstream text("namen.txt");
    text.imbue(unicode);
    vector<wstring> v;
    wstring line;
    while( getline(text,line) ) {
        v.push_back(line);
    }
}
```

(\_\_ / 15)

3. Durchlaufen Sie den in Aufgabe 2 erzeugten Container mit  
a) einem Iterator und geben sie die Werte auf dem Terminal aus.

```
//Einbinden
#include<iostream>
// In der main
vector<wstring>::iterator it;

for(it = v.begin(); it != v.end(); it++) {
    wcout << * it << endl;
}
```

- b) der for\_each-Funktion aus der STL und geben die Länge jedes Eintrags auf dem Terminal aus.

```
//Einbinden
#include<algorithm>
// Neue Funktion
void doPrint(wstring elem) {
    wcout << elem.length() << endl;
}
// In main
for_each(v.begin(),v.end(), doPrint);
```

(\_\_ / 10)

---

4. Schreiben sie eine Klasse "Buch". Mit den privaten Attributen "wstring titel" ,"int seitenzahl" und "bool gelesen". Eine öffentliche Methode "lesen" soll den Wert von gelesen auf "true" setzen. "titel" und "seitenzahl" sollen über den Konstruktor festgelegt werden.

a) Header der Klasse "Buch"

```
#ifndef BUCH_H
#define BUCH_H
#include<string>
class Buch {
public:
    Buch(std::wstring, int);
    void lesen();
private:
    std::wstring titel;
    int seitenzahl;
    bool gelesen;
};
#endif /* BUCH_H */
```

b) Implementierungsdatei der Klasse "Buch"

```
#include "Buch.h"
using namespace std;
Buch::Buch(wstring title, int seitenzahl) {
    this -> titel = title;
    this -> seitenzahl = seitenzahl;
    this -> gelesen = false;
}

void Buch::lesen() {
    this->gelesen = true;
}
```

c) main-Methode in der zwei Bücher erzeugt werden und eines davon gelesen wird.

```
#include "Buch.h"
int main() {
    Buch gutesBuch(L"Die unerträgliche Leichtigkeit des Seins",300);
    Buch schwachesBuch(L"Der Steppenwolf",229);
    gutesBuch.lesen();
}
```

5. Schreiben sie ein Programm, dass die Zahlen von 1 bis 10 000 auf das Terminal ausgibt.

Bei allen:

```
#include <iostream>
using namespace std;
int main() { a) | b) | c) }
```

- a) Mit einer while-Schleife.

```
int i;
i = 1;
while (i <= 10000 ) {
    cout << i << endl;
    i += 1;
}
```

- b) Mit einer for-Schleife.

```
int i;
for (i = 1 ; i <= 10000; i++ ) {
    cout << i << endl;
}
```

- c) Mit einer do-while-Schleife.

```
int i;
i = 0;
do {
    cout << ++i << endl;
}
while (i < 10000);
```

(\_\_ / 10)

6. Beschreiben sie die verschiedenen Lebensphasen der String-Objekte

```
#include <string>
int main() {
    int i = 10;
    std::wstring eineInstanz; // Deklaration, Speicher wird bereitgestellt
    eineInstanz = L"Etwas Text"; // Zuweisung eines Werts

    while (i >= 0 ) {
        std::string x("" + i); // Instanziierung: Konstruktor wird bei jedem
        // Schleifendurchlauf aufgerufen, danach Destruktor
    }
    // beim Verlassen der main() wird der Destruktor von „eine Instanz aufgerufen“
}
```

(\_\_ / 5)

---

7. Finden sie die logischen Fehler in den Quellcodefragmenten, korriegieren sie diese, und erklären sie deren Auswirkung.

| Quelltext  | Korrektur  |
|--|--|
| <pre>#include&lt;string&gt;  int main(void) {     string s = "test";     cout &lt;&lt; s &lt;&lt; endl; }</pre>  | <pre>#include&lt;iostream&gt;  using namespace std;</pre>  |
| <p>Fehlerbeschreibung:<br/>Bibliothek für Ein- und Ausgabe nicht eingebunden.<br/>Kein Zugriff auf string, cout und endl da sie im Namensraums std liegen.<br/>Programm wird nicht kompilieren.</p>  | <p>Was korrigiert und warum:<br/>Ein- und Ausgabebibliothek eingebunden.<br/>Namensraum std zum Namensraum des Programms hinzugefügt.</p>  |
| <pre>using namespace std; int main() {      int i = 0;     bool flag = false;     while (flag = true) {         cout &lt;&lt; i &lt;&lt; " ist kleiner als 3" &lt;&lt; endl;         flag = lessThanThree(i);         i++;     } }</pre>                 | <pre>#include &lt;iostream&gt;  // vor main bool lessThanThree(int x) { return x &lt; 3; } while(flag) {}</pre>  |
| <p>Fehlerbeschreibung:<br/>Bedingung in der while Schleife bleibt stets war.<br/>Programm wird nicht terminieren.<br/><br/>Bibliothek für Ein- und Ausgabe nicht eingebunden.<br/>Funktion lessThanThree fehlt.<br/>Programm wird nicht kompilieren.</p> | <p>Was korrigiert und warum:<br/>Bibliothek iostream eingebunden damit Ein- und Ausgabe möglich.<br/>Funktion lessThanThree implementiert.<br/>Bedingung in while-Schleife ohne =lessThanThree da Vergleichsoperator == und die Funktion bereits bool zurückgibt, was für den Test der Schleife ausreicht.</p> |

| Quelltext   | Korrektur   |
|---|---|
| <pre>using namespace std; int main() {     setlocale(LC_ALL, "de_DE.UTF-8");     vector&lt;int&gt; v;     string s;     while( getline(wcin,s) ) {         v.push_back(s);         wcout &lt;&lt; s &lt;&lt; " hat die Länge" &lt;&lt; s.length() &lt;&lt; endl;     } }</pre>                                      | <pre>#include&lt;vector&gt; #include&lt;iostream&gt; #include &lt;string&gt; wstring s;</pre>   |
| <p>Fehlerbeschreibung</p> <p>Bibliotheken für Vektor und Ein- und Ausgabe nicht eingebunden.<br/>Programm wird nicht kompilieren.<br/>String statt wstring verwandt. Falsche Längenangabe.</p>  | <p>Was korrigiert und warum:</p> <p>Bibliotheken eingebunden damit es kompiliert.<br/>wstring statt string, da sonst length() das falsche Ergebnis liefert.</p> |
| <pre>#include&lt;vector&gt; #include&lt;iostream&gt; int main() {     std::vector&lt;int&gt; v(10);     int i=10;     while (i &gt;= 0) {         v.push_back(i);     }     std::vector::iterator it;     for (it = v.begin(); it != v.end(); v++) {         std::cout &lt;&lt; i &lt;&lt; std::endl;     } }</pre> | <p>In der while-Schleife:</p> <pre>v.push_back(i); i--;</pre> <p>In der for-Schleife:</p> <pre>std::cout &lt;&lt; * i &lt;&lt; std::endl;</pre>                 |
| <p>Fehlerbeschreibung</p> <p>Programm terminiert nicht.</p> <p>Es wird die Adresse, nicht der Zahlenwert beim durchlaufen des Vektor ausgegeben.</p>  | <p>Was korrigiert und warum:</p> <p>Zählervariable wird bei Schleifendurchlauf nun inkrementiert.<br/>Iterator wird nun dereferenziert.</p>                     |

## 8. Finden sie die Syntaxfehler

a)

```
/ Permutationen                                !
#include <iostream>                               !
#include <string>
#include <algorithm>
using namespace std;

int main() {
    wstring satz[] = (L"Max",L"liebt,L"Maria");      !

    wcout >> "The 3! possible permutations with 3 elements:" >> endl;      !!

    sort (satz; satz+3)                            !!

    do {
        wcout << satz.at[0] << " " << satz.at[1] << " " << satz.at[2] << endl;      !!!
    } while ( next_permutation, (satz,satz + 3)) );    !!

}
```

b)

```
/* Header der Klasse Tier /*                      !
#ifndef TIER_H_                                     !
#define TIER_H_                                     !
#include <string>
#include <iostream>

class Tier {
    private:
        std::string geschlecht                      !

        protected                                     !
            std::string laut;

        public:
            Tier( string );
            std::string name;
};

#endif
```

(\_\_ / 15)

---