

Fachhochschule Regensburg

Test Nr. 8	Testteilnehmer
Programmieren in C++ Aufgabensteller: Prof. Sauer	Name: _____ Vorname: _____
Juni 2005 Arbeitszeit: 45 Minuten	Zugel. Hilfsmittel: Skriptum

1. Aufgabe

1. Für einfache Anwendungen, wie z. B. einer Funktion zum Quadrieren von Zahlen, könnte man auf die Idee kommen, den Präprozessor zu missbrauchen, um die Funktion für verschiedene Typen zur Verfügung zu stellen. Statt also für jeden benötigten Typ eine Funktion zu schreiben, z.B.

```
inline int quad(int x) {return x * x; }  
inline short quad(short x) {return x * x; }  
inline double quad(double x) {return x * x; }
```

schreibt man ein kurzes Makro

```
#define quad(a) (a) * (a)
```

, das dann aber sofort den großen Nachteil von Makros aufzeigt:

In der folgenden Anweisung läuft dann auch Einiges fehlerhaft.

Schreibe in den angegebenen Kommentarzeilen auf, wie der Makro aufgelöst wird und welcher Wert der jeweiligen Variable zugeordnet wird. Außerdem lege fest, ob hier ein Fehler aufgetreten ist oder alles „ok“ verlaufen ist.

```
#include <iostream.h>  
  
#define quad(a) (a) * (a)  
  
int main()  
{  
    int x = 3, y;  
    short m = 4, n;  
    double a = 4.5, b;  
  
    y = quad(x); //  
  
    n = quad(m); //  
  
    b = quad(a); //  
  
    y = quad(x++); //  
    //  
  
    cout << "Wert von y = quad(x++): " << y << endl; //  
    cout << "Wert von x          : " << x << endl; //  
  
}
```

2. Ersetze den vorliegenden Makro durch eine Funktionsschablone

3. Das unter 1 vorliegende Hauptprogramm soll mit dem unter 2. angegebenen Funktionstemplate arbeiten. Gib das Hauptprogramm an, das mit dem unter 1. angegebenen Funktionstemplate zusammenarbeitet.

4. Die folgende Funktionsschablone

```
template <class T> inline T max(T a, T b) { return a > b ? a : b; }
```

kann durch eine der Anweisungen im nachfolgenden Hauptprogramm nicht überschrieben werden.

```
int main()
{
    int x = 3, y = 3, z = 4;
    double a, b = 1.4, c = 7.3;
    short m, n = 6, o = 9;

    x = max(y, z); //
    a = max(b, c); //
    m = max(n, o); //
    x = max(y, o); //
}
```

Markiere im Zeilenkommentar, wo der Fehler auftritt!

5. Wie könnte der Fehler durch eine explizite Typumwandlung beim Aufruf behoben werden?

```
x = max(y, (int) o); // Explizite Typumwandlung
```

6. Ist die folgende Definition zulässig?

```
template <class T1, class T2, class T3>
    inline T1 max(T2 a, T3 b) { return a > b ? a : b; }
```

7. Warum ergibt ein Aufruf für die unter 4. angegebene Maximumfunktion für C-Strings keinen Sinn?

8. Wie würde eine korrekt formulierte Maximumfunktion für Strings lauten?

2. Aufgabe

Gegeben sind folgende Templatefunktionen

```
template <class T>
inline T f(T a, T b) { }    // #1

template <class T1, class T2>
inline T1 f(T1 a, T2 b) { }    // #2

template <class T>
inline T* f(T* a, T b) { }    // #3

template <class T>
inline T f(templateT a, T* b) { }    // #4

template <class T>
inline T f(T* a, T b) { }    // #5

template <class T>
inline T* f(T a, T b, int c) { } // #6
```

1. Welche Kombinationen der hier angegebenen Templatefunktionen sind zulässig (Maximalauswahl)

2. Welche Kombinationen sind wegen Mehrdeutigkeiten nicht zulässig? (Minimalauswahl unzulässiger Funktionen)
