

Fachhochschule Regensburg

Test Nr. 7	Testteilnehmer
Programmieren in C++ Aufgabensteller: Prof. Sauer	Name: _____ Vorname: _____
Mai 2006 Arbeitszeit: 45 Minuten	Zugel. Hilfsmittel: Skriptum

1. Aufgabe

Gegeben ist das folgende Programm zur Manipulation von Zeigern:

```
#include <iostream.h>
int main()
{
    int a=5; // _____

    cout<<a<<endl; // _____

    int *pa=&a; // _____

    *pa=6; // _____

    cout<<a<<endl; // _____

    a=7; // _____

    cout<<*pa<<endl; // _____

    int b=8; // _____

    int *pb=pa; // _____

    cout<<*pb<<endl; // _____

    int **ppa=&pa; // _____

    cout<<**ppa<<endl; // _____

    ppa=&pb; // _____

    cout<<**ppa<<endl; // _____

    pb=&b; // _____

    cout<<**ppa<<endl; // _____

};
```

Jede Zeile des Programms enthält innerhalb von `main()` ein Kommentarzeichen `“//“`.

a) Gib nach `//` einen Kommentar an, der beschreibt, was die in derselben Zeile stehende Anweisung bewirkt.

b) Übertrage den Quelltext in eine Datei und übersetze das Programm. Kontrolliere, inwieweit ihre Kommentare dem wahren Sachverhalt entsprechen.

2. Aufgabe

Gegeben ist das folgende Programm:

```
#include <iostream.h>

struct Datum
{
    int Monat;
    int Tag;
    int Jahr;
};

int main()
{
    int index, *zeiger1, *zeiger2;
    zeiger1 = &index;
    *zeiger1 = 77;
    zeiger2 = new int;
    *zeiger2 = 173;
    cout << "Die Werte sind " << index << " "
         << *zeiger1 << " " << *zeiger2 << endl;
    /*
    */
    zeiger1 = new int;
    zeiger2 = zeiger1;
    *zeiger1 = 999;
    cout << "Die Werte sind " << index << " "
         << *zeiger1 << " " << *zeiger2 << "\n";
    /*
    */
    delete zeiger1;
    float *floatZeiger1, *floatZeiger2 = new float;
    floatZeiger1 = new float;
    *floatZeiger2 = 3.14159;
    *floatZeiger1 = 2.4 * (*floatZeiger2);
    delete floatZeiger2;
    delete floatZeiger1;
    Datum *DatumZeiger;
    DatumZeiger = new Datum;
    DatumZeiger->Monat = 11;
    DatumZeiger->Tag = 13;
    DatumZeiger->Jahr = 1940;
    cout << DatumZeiger->Monat << "/" << DatumZeiger->Tag << "/"
         << DatumZeiger->Jahr << "\n";
    /*
    */
    delete DatumZeiger;
    char *cZeiger;
    cZeiger = new char[37];
    delete [] cZeiger;
    cZeiger = new char[sizeof(Datum) + 133];
    delete [] cZeiger;
    return 0;
}
```

- Gib (in den Kommentarzeilen) an, welche Ausgabe die unmittelbar vorher stehende Ausgabeanweisung erzeugt.
- Überprüfe durch Implementierung und Test des vorliegenden Programms, ob die Angaben korrekt sind.

3. Aufgabe

Gegeben ist das folgende Programm

```
#include <iostream.h>

void druckeEtwas(float zuIgnorierendeDaten);
void druckeNachricht(float aufzulistendeDaten);
void druckeFloat(float zuDruckendeDaten);
void (*funktionsZeiger)(float);

int main()
{
    float Pi = 3.14159;
    float ZweiPi = 2.0 * Pi;
    druckeEtwas(Pi);
    funktionsZeiger = druckeEtwas;
    funktionsZeiger(Pi);
    funktionsZeiger = druckeNachricht;
    funktionsZeiger(ZweiPi);
    funktionsZeiger(13.0);
    funktionsZeiger = druckeFloat;
    funktionsZeiger(Pi);
    druckeFloat(Pi);
    return 0;
}

void druckeEtwas(float zuIgnorierendeDaten)
{
    cout << "Das ist die Funktion DruckeEtwas" << endl;
}

void druckeNachricht(float aufzulistendeDaten)
{
    cout << "Die aufzulistenden Daten sind: "
        << aufzulistendeDaten << endl;
}

void druckeFloat(float zuDruckendeDaten)
{
    cout << "Die zu druckenden Daten sind: "
        << zuDruckendeDaten << endl;
}
```

a) Welche Ausgaben erzeugt dieses Programm?

b) Überprüfe die unter a) gemachten Angaben durch Implementierung und Test

4. Aufgabe

Welche Ausgaben erzeugt das folgende Programm?

```
#include <iostream.h>

void f(int* a)
{
    cout << * a << endl;
}

void g(int a)
{
    cout << a << endl;
}
```

```

int main()
{
    int a = 3, b = 4;
    int* pa = &a;
    f(pa);      // Pointeruebergabe
    f(&b);      // Adresse eines int uebergeben
    g(a);       // int uebergeben
    g(*pa);     // Pointer erst dereferenzieren
}

```

5. Aufgabe

Welche Anweisungen in dem folgenden Programm sind falsch?

```

int main()
{
    char*
        t1 = "Das ist ein voellig veraenderbarer String";
        const char*
        t2 = "Inhalt ist nicht veränderbar, wohl aber der Pointer";
    char* const
        t3 = "Dieser Pointer ist konstant, nicht aber der Inhalt";
        const char* const
        t4 = "Pointer und Inhalt sind nicht veraenderbar";
    char * s = "Noch ein String";

    *t1 = 'z';
    *t2 = 'z';    // _____
    *t3 = 'z';
    *t4 = 'z';    // _____

    t1 = s;
    t2 = s;
    t3 = s;       // _____
    t4 = s;       // _____

    s = t2;
    // Warnung: Umwandlung const char* -> char*
    // vernichtet const-Eigenschaft
    s = (char*) t2; // _____
}

```
