

2. Semester, 2. Prüfung, Version 2

Name	
------	--

- Die gesamte Prüfung bezieht sich auf die Programmierung in C++!
- Prüfungsdauer: 90 Minuten
- Mit Kugelschreiber oder Tinte schreiben
- Lösungen können direkt auf die Aufgabenblätter geschrieben werden
- PC's sind nicht erlaubt
- Unterlagen und Bücher sind erlaubt
- Achte auf Details wie Punkte, Kommas und Semikolons

Aufgabe	Punkte	
Vektoren/Arrays	10	
Teilobjekte und statische Elemente	12	
Programmverständnis	8	
Zeiger und Vektoren	10	
Weitere Fragen	10	
<i>Total</i>	50	

1. Vektoren/Arrays

- a) Wieviele Bytes an Speicher werden für folgenden C-String gebraucht?
(1 Punkt)

```
char text[] = „Viel Spass mit C++“;
```

- b) Erzeuge ein Array von 58 doubles mit dem Namen *datenarray*. Du brauchst das Array nicht zu initialisieren. (1 Punkt)

- c) Definiere ein Array von 100 char-Werten und schreibe eine Schleife, die alle Werte mit einem 'A' initialisiert (2 Punkte)

- d) Definiere ein Array von 10 long-Werten und schreibe eine Schleife, die alle Werte mit **33** initialisiert. (2 Punkte)
- e) Definiere ein Array von 333 unsigned long - Werten und initialisiere es auf einer Zeile so, dass alle Elemente den Wert 0 haben. (2 Punkte)
- f) Was gibt folgender Code auf dem Bildschirm aus? (2 Punkte)

```
#include <iostream>
using namespace std;
int main()
{
    char text[] = { 'H', 'a', 'h', '?', 0 };
    cout << text << endl;
    return 0;
}
```

2. Teilobjekte und statische Elemente

- a) Definiere zuerst eine Klasse *GezaehlttesObjekt*. (1 Punkt)
Die Klasse kann nichts, sie wird aber die Anzahl Objekte zählen, die von ihr erzeugt werden (Konstruktor) und wenn ein Objekt zerstört wird (Destruktor), soll der Zähler erniedrigt werden.
Diese Klasse soll also einen Konstruktor (ohne Parameter) und einen Destruktor haben. (1 Punkt)
Die Klasse soll auch eine statische Methode *holeAnzahlObjekte* haben. (1 Punkt)
Die Anzahl Objekte wird in einer statischen Variable gespeichert. (1 Punkt)

Schreibe danach direkt unter die Klassendefinition den Code, der die statische Zählvariable initialisiert, (1 Punkt) sowie den Code für den Konstruktor und den Destruktor. (2 Punkte)
(Total 7 Punkte)

- b) Gegeben ist die Definition der Klasse *Adresse*, mit einem Konstruktor mit Parametern. Gegeben ist auch die Klasse *Person*, die eine Adresse, einen Vornamen und Nachnamen als Datenelemente enthält. Schreibe den Code für den Konstruktor der Klasse *Adresse*. (2 Punkte) Schreibe darunter den Code für den Konstruktor der Klasse *Person*. (3 Punkte)
(Total 5 Punkte)

```
#include <string>
using std::string;
class Adresse
{
    public:
        Adresse(const string& Strasse,
                const string& Ort);
    private:
        string m_Strasse;
        string m_Ort;
};

class Person
{
    public:
        Person(const string& Vorname,
               const string& Nachname,
               const string& Strasse,
               const string& Ort);
    private:
        Adresse m_Adresse;
        const string m_Vorname;
        const string m_Nachname;
};

// Hier Konstruktor für Adresse und darunter
// Konstruktor von Person
```

3. Programmverständnis

- a) Was gibt folgendes Programm aus? Gehe die Schleife Schritt für Schritt durch und schreibe für jeden Schleifendurchgang die Werte der Variablen auf. (4 Punkte)

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

int main()
{
    char buchstaben[] = { 'h','e','r','a','l','o',' ' };

    int indices[10] = { 0, 1, 2, 2, 6, 0, 3, 4, 4, 5 };

    for(int i = 0; i < 10; ++i)
    {
        int index = indices[i];
        char buchstabe = buchstaben[index];
        cout << buchstabe;
    }
    return 0;
}
```

- b) Was gibt das Programm auf der folgenden Seite aus? (4 Punkte)

```
#include <iostream>
#include <string>
using namespace std;

enum Sprache
{
    Englisch,
    Deutsch
};

char* Texte[] =
{
    "Is cool man !",
    "Isch kuul maen !"
};

class CoolDinky
{
public:
    CoolDinky(const string& name); // Konstruktor
    void setSprache(Sprache dieSprache);
    void sprich() const;
private:
    Sprache m_sprache;
    string m_name;
};

CoolDinky::CoolDinky(const string& name)
    :m_name(name), m_sprache(Englisch)
{
}

void CoolDinky::setSprache(Sprache dieSprache)
{
    m_sprache = dieSprache;
}

void CoolDinky::sprich() const
{
    cout << m_name << " sagt : ";
    cout << Texte[m_sprache] << endl;
}

// Hier ist das main
int main()
{
    CoolDinky dinkies[] =
    {
        "Hank",
        "John"
    };

    dinkies[1].setSprache(Deutsch);
    for(int i = 0; i < 2; ++i)
    {
        dinkies[i].sprich();
    }

    return 0;
}
```

4. Zeiger und Vektoren

a) Was gibt folgendes Programm aus? (1 Punkt)

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

int main()
{
    char* cstring = "Hallo";
    cout << cstring << endl;

    return 0;
}
```

b) Was gibt folgendes Programm aus? (2 Punkte)

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    char text[] = "Nicht luegen ist gut";
    char* ausgabe = text + 6;
    cout << ausgabe << endl;

    return 0;
}
```

c) Was hat die Variable z hier für einen Wert? (2 Punkte)

```
int main()
{
    char text[] = "Nicht luegen ist gut";
    int z = &text[6] - &text[3];

    return 0;
}
```

d) Was hat die Variable z hier für einen Wert? (2 Punkte)

```
int main()
{
    int zahlen[] = { 1, 2, 3, 4, 5};
    int* pz = zahlen + 2;
    int z = *pz;

    return 0;
}
```

e) Was gibt folgendes Programm aus? (3 Punkte)

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

int main()
{
    char* cstring = "zzksfsjiereslkdnvjfp";
    char* p = &cstring[10];

    cout << p << endl;

    int i = 0;
    while(i < 20)
    {
        cout << cstring[i];
        i = i + 2;
    }

    return 0;
}
```

5. Weitere Fragen

- a) Definiere eine Aufzählung *Einstufung* für eine Beurteilung, so dass gilt:
miserabel = 2, ungenuegend = 3, genuegend = 4, gut = 5,
hervorragend = 6
(2 Punkte)
- b) Eine IP-Adresse wie sie in Netzwerken zur Identifikation von Geräten (meistens PC's) verwendet wird, steht häufig in folgender Form:
192.168.0.255
Wobei die einzelnen Zahlen jeweils einen Wert zwischen 0 und 255 haben können. Definiere eine Struktur, die die vier einzelnen Werte (A, B, C, D) als Elemente enthält. Verwende nicht mehr Speicher als nötig! (3 Punkte)
- c) Was hat die Variable *b* für einen Wert? (2 Punkte)
- ```
int x = 10;
int y = 11;
int& z = y;
z--;
bool b = (x == y);
```
- d) Was hat die Variable *x* für einen Wert? (2 Punkte)
- ```
int x = 10;  
int* py = &x;  
(*py)++;
```
- e) Willst Du fehlerfreie Programme schreiben? (1 Punkt)