3. Semester : 1. Prüfung

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
- PCs sind nicht erlaubt
- Unterlagen und Bücher sind erlaubt
- Die Aufgaben sind in Fragen unterteilt, die mit Kleinbuchstaben gekennzeichnet sind
- Achte auch auf Details wie Punkte oder Kommas und Semikolons. !!!!!!
- In den Beispielen fehlt meistens folgender Code, der als gegeben gilt:

```
#include <iostream>
using namespace std;
int main()
{
....
    return 0;
}
```

Aufgabe	**Punkte**	
TOTAL		

1. Klassen mit Datenelementen

a) Konstante Datenelemente einer Klasse müssen auf besondere Art initialisiert werden! Gegeben sei die folgende Klasse “Kreis”. Schreibe den Konstruktor für diese Klasse direkt unter die Klassendeklaration (4 Punkte)

```
class Kreis
{
    public:
        Kreis();
    private:
        const double pi;
};
```

b) Schreibe den Konstruktor für folgende Klasse, so dass alle Datenelemente, die es nötig haben sinnvoll initialisiert werden (2 Punkte)

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

class PersonenDaten
{
    public:
        PersonenDaten();
    private:
        string m_Name;
        double m_Gewicht;
};
```

2. dynamischer Speicher

a) Wieviel Bytes Speicher werden im folgenden Beispiel alloziert ? (1 Punkt)

```
long* pLong = new long[20];
```

b) Schreibe im folgenden Beispiel die Zeile Code auf, die es benötigt um den Speicher wieder freizugeben, der alloziert wird (2 Punkte).

```
long* pLong = new long[20];
```

c) Schreibe wie im Beispiel b) den Code um den Speicher freizugeben, der alloziert wird (2 Punkte).

```
long* pLong = new long;
```

3. Klassen und dynamischer Speicher

a) Betrachte folgende Klassendeklaration und -definition und die main-Funktion darunter. Wieviel Speicher wird am Ende nicht richtig freigegeben ? (2 Punkte)

```
class Leck
{
    public:
        Leck();
        ~Leck();
    private:
        char* m_data;
};

Leck::Leck()
{
    m_data = new char[4];
}

Leck::~Leck()
{
}

int main()
{
    Leck leckEins;
    Leck leckZwei;
    return 0;
}
```

b) Was muss geändert werden, damit der Speicher richtig freigegeben wird ? (2 Punkte)

c) Schreibe den korrigierten oder ergänzten Code nieder (nur was geändert werden muss oder die Funktion, die geändert werden muss) (2 Punkte)

d) Was gibt folgendes Programm aus ? (4 Punkte)

```
class WinkyDinky
{
    public:
        WinkyDinky();
        ~WinkyDinky();

        void SagHallo();
};

WinkyDinky::WinkyDinky()
{
    cout << “Ich bin WinkyDinky” << endl;
}

WinkyDinky::~WinkyDinky()
{
    cout << “Ich gehe jetzt” << endl;
}

void WinkyDinky::SagHallo()
{
    cout << “Ich sag Hallo” << endl;
}

int main()
{
    WinkyDinky* winky = new WinkyDinky;
    winky->SagHallo();
    return 0;
}
```

4. Weitere Fragen (auch zu Klassen)

a) Im Diagramm unten haben wir die zwei Klassen Auto und Sportwagen. Die verwendete Notation zeigt an, dass die Klasse Sportwagen von der Klasse Auto abgeleitet ist. Der Pfeil zeigt dabei von der abgeleiteten Klasse zur Elternklasse.
Ergänze nun das Diagramm mit folgenden Klassen und setze die Klassen in die richtige Beziehung zueinander indem du herausfindest welche Klassen voneinander ableiten. Es kann sein, dass eine Klasse überhaupt nicht in Beziehung zu einer anderen Klasse steht. Zeichne diese einfach alleine auf. Klasse “Fortbewegungsmittel”, Klasse “Fahrrad”, Klasse Ferrari”, Klasse “Kombi”, Klasse “Person”. (6 Punkte)