

Es sind die vorgegebenen Datentypen zu beachten (int Werte werden in Variable vom Typ int eingelesen und double Werte in Variable vom Typ double).

Keine Zeichenketten oder Strings verwenden, wenn Zahlen verlangt sind.

Wenn ein Ergebnis berechnet werden soll, dann reicht die Ausgabe alleine (außer wenn nur diese explizit verlangt ist) nicht aus. Der Wert muss in einer Variable zur Verfügung stehen, sodass er prinzipiell für weitere Berechnungen verwendbar wäre.

Verwendung von globalen Variablen und goto ist (soweit sie nicht in der Angabe explizit empfohlen wird) untersagt.

Achten Sie darauf, dass die Objekte Ihrer Klasse in jedem Fall in einem konsistenten Zustand sein müssen.

Erweitern Sie die Klasse Person (Evaluierungsaufgabe der Vorwoche) und schreiben Sie einen Kopierkonstruktor und einen Kopierzuzuweisungsoperator, die jeweils den dynamisch allozierten Speicher ordnungsgemäß kopieren.

Schreiben Sie weiters einen Ausgabeoperator << für die Klasse Person, der die gespeicherten Werte in folgendem Format ausgibt:

[Person: name] z.B. [Person: Peter Rosegger]

Die Angabe der Vorwoche:

Implementieren Sie eine Klasse Person mit einer Instanzvariable name. Der Name einer Person soll in einem dynamisch allozierten char Array gespeichert werden (die Länge des Arrays kann als konstant - z.B. 80 Zeichen – angenommen werden).

Schreiben Sie einen Konstruktor, der eine Zeichenkette (char-Array) als Parameter erhält und den Namen entsprechend setzt.

Schreiben Sie eine Methode void taufen(const char \*), die einer Person einen neuen Namen zuweist. Beachten Sie, dass die Maximallänge des Arrays nicht überschritten werden darf. Ein dynamisches Wachstum des Arrays ist nicht notwendig.

Schreiben Sie einen Destruktor, der den dynamisch allozierten Speicher wieder ordnungsgemäß freigibt.

Zusatzpunkt (30 Minuten zusätzlich) :

Schreiben Sie eine operator[] Methode, die in einem dynamisch allozierten Array eine Zeichenkette (char \*) retourniert, die genau jene Zeichen enthält, deren Position im Namen ein ganzzahliges Vielfaches des gegebenen Indexwertes ist.

```
Person p("Peter Rosegger");  
cout << p[3]; //soll PeRee ausgeben.
```

Testen Sie Ihre Klasse ausgiebig.

Eine mögliche Lösung:

```
#include <iostream>  
using namespace std;  
  
const int namelen=80;  
  
class Person {  
    char *name;  
public:  
    explicit Person(const char *name): name(new char[namelen+1]) {  
        int i=0;  
        for (; i<namelen && name[i]; ++i)  
            this->name[i]=name[i];  
        this->name[i]='\0';  
    }  
    ~Person() {  
        delete[] name;  
    }  
    void taufen(const char *neu) {  
        int i=0;  
        for (; neu[i] && i<namelen; ++i)  
            name[i]=neu[i];  
        name[i]='\0';  
    }  
    void print() {  
        cout<<name;  
    }  
}
```

```
char *operator[] (int ind) {
    char *res = new char[namelen+1];
    int pos=0;
    for (int i=0; name[i]; i++)
        if (i%ind == 0)
            res[pos++]=name[i];
    res[pos]='\0';
    return res;
}
};
```