

Kreuze an, ... falsches kreuz → minuspunkte....

	Darf im Header sein	MUSS in der .c sein
Struct {int l, char e};		
Struct ss s;		
Struct s;		
Extern struct ...		

vererbung ankreuzen:

was bekommen c++ und java klassen bei der vererbung ?

funktionsprototypen
deklarationen
definitionen

was bekommt c++ nach der vererbung was java nicht bekommt?

...

Wie kann man ambiguous Verhalten erzeugen (class.x soll ambiguous sein)

so wenig code wie möglich schreiben

```
class A { /*leere klasse */ }
```

```
A a;
```

```
a.x = 1;
```

Kreuze an → falsches kreuz → minuspunkt

```
int main(void){
```

```
    int l = -;
```

```
    int* p = (int*)malloc(sizeof(int));
```

```
}
```

	richtig	falsch
l und *p liegen grundsätzlich gebeneinander im stack		
P zeigt auf eine adresse im heap		
.... +3 weitere fragen		

Stackframe aufschreiben bei */stop/* (7/8 Punkte)

```
typedef struct {
    int n;
    char c;
} data;

void foo(data *dp, data d1) {
    char v;
    dp--;
    v=dp->c;
    d1.n=5;
    /*stop*/
}

int main(void) {
    data d[] = {{1, 'a'}, {3, 'b'}};
    char v = 'A';
    foo(d+1, d[1]);
    return 0;
}
```

Entscheide ob richtig oder falsch

```
class Object{ // in der klausur war object = "A "  
    private:  
        string a;  
    public:  
        Ort();  
};
```

```
>- const Objekt& func() {  
    Objekt neu;  
    return neu;  
}
```

- 1) fehler da rückgabetyt unveränderbar ist
- 2) fehler da Object eine genaue kopie von a erstellt
- 3 fehler da ein null string zurückgegeben wird als const.
4. kein compilerfehler, unerwartetes verhalten
5. ?

Vererbung

Person:

soll die Fertige Klasse (von ihm vorgegeben) Ort benutzen . Wohnort und Geburtsort sollen Zeiger sein.

Person soll ein konstruktor aus 4 attributen haben, Wohnort, Geburtsort, Name, alter
Eine Setter Methode
und ein Destruktor.

Baby:

Konstruktor mit drei attributen haben, Name wohnort geburtsort,

Die alter setter Methode soll umgeschrieben werden. Falls das Alter beim setzen !=0 ist, soll eine exception geworfen werden. Klasse <stdexception> benutzen. (es wurde ein excpetiontyp angegeben). Dieser nimmt nach Konstruktorauf Ruf ein String an.

Alle methoden Funktionen sollen Falls möglich Konstantiert werden.

Keine extraMethoden erlaubt.

Seperat in .cpp .h trennen. Defines und includes angeben!

...

3 Weitere Kriterien angegeben ,

+ Geschriebener Code soll in der unteren Main klasse ausführbar sein.

```
int main(){
    Ort hannover {"Hannover", 10 , 10};
    Ort hamburg {"Hamburg", 10 , 10};

    Person p = {&hannover, &hamburg, 10, "John doe"};
    Baby b = {&hamburg, &hannover, "Ann-Christin"};

    p.setAlter(26);
    b.setAlter(3); // soll fehler werfen

    cout << b.getName() << endl;
    return 0;
}
```

Operatoren
überladen

6 Operatoren überladen (13 Punkte)

schreiben Sie den

- copy Konstruktor (tiefe kopie)
- operator =
- operator == (breite laenge wie auch die *zeichenkette*(inhalt) vom namen sollen gleich sein)

vermeiden Sie Redudanz und keine weitem methoden

- wenn kein name gegeben dann ist es ein nullptr

Ort.h

```
class Ort {  
private:  
    int leange;  
    int breite;  
    std::string* name;  
public:  
    // hier der code  
}  
  
int main() {  
    Ort o1 {"Hannover", 10 , 10};  
    Ort o2 {"Hamburg", 10 , 10};  
}
```

C – Liste kopieren

erstelle eine tiefe Kopie der liste , nur durch angabe des heades.

Vollende die funktion

Knoten soll ein struct sein und hat ein int wert und ein zeiger von typen Knoten namens "next".

Leere Liste soll auch beachtet werden!

Knoten* copy(knoten* head){

}