

Hochschule Deggendorf Dr. Peter Jüttner	
Vorlesung: Einführung in die Programmierung	WS 2012
Übung 13	Termin 15.1.13

## Pointer (Zeiger)

Bearbeiten Sie die folgenden Aufgaben in Gruppen

Theorie

### A1. Theorie

1. Welchen Typ von Werten speichern Pointervariable?
2. Wie kann man sich eine Pointervariable „bildlich“ vorstellen?
3. Wie wird eine Pointervariable deklariert? Geben Sie ein Beispiel
4. Wie wird ein Wert an eine Pointervariable zugewiesen?
5. Welche Werte werden sinnvollerweise an Pointervariable zugewiesen?
6. Was versteht man unter Dereferenzieren? Wie wird eine Pointervariable dereferenziert
7. In folgendem Programmstück werden Variable ivar und ivar2 vom Typ Integer deklariert und Pointervariable ip und ip1 vom Typ Pointer auf Integer. Danach werden einige Zuweisungen ausgeführt. Nehmen Sie an die Variable ivar hat die Adresse 0x12345678 und ivar2 hat die Adresser 0x11223344  
Welchen Wert haben die Variablen ip und ivar nach jeder Zuweisung?  
Stellen Sie die Zuweisungen bildlich dar

```
int *ip = 0;
int ivar = 10;
ip = &ivar;
*ip = 20;
int ivar2 = 100;
int *ip2 = ip1;
ip1 = &ivar2;
*ip1 = 77;
```

8. Was bedeutet es, wenn ein Pointer vom Typ void ist?

9. Wie kann ein void-Pointer in einen anderen Pointer umgewandelt werden?
10. Was ist der Heap Speicherbereich?
11. Wie wird mit dem Heap gearbeitet? (Speicher anfordern und freigeben)

## **A2. Praxis**

1. Programmieren Sie die Codesequenz aus A1.7. Geben Sie dabei die Werte der Variablen nach jedem Schritt aus.
2. Schreiben Sie ein Prozedur Pointerausgabe mit einem Parameter vom Typ void\*. Die Prozedur soll den Wert des Pointers ausgeben und den Inhalt des Speicherbereichs, auf den der Parameter zeigt, jeweils interpretiert als char, short, int unsigned int und float. Casten Sie dazu den Pointer auf den Typ bevor sie ihn dereferenzieren  
Rufen Sie die Prozedur mit den Adressen entsprechender Variablen der angegebenen Typen auf.
3. Schreiben Sie eine Prozedur, die über die Adressen dreier Variablen deren Werte tauscht. Dabei soll die erste Variable den Wert der zweiten, die zweite den Wert der dritten und die dritte den Wert der ersten Variablen erhalten.
4. Schreiben Sie ein C-Programm, das auf dem Heap Speicher für eine Zahl Typ int reserviert. Schreiben Sie mittels scanf(...) einen Wert direkt über die Adresse in den Speicher, geben Sie den Wert mittels printf wieder aus. Danach geben Sie den Speicher wieder frei.
5. Schreiben Sie ein C-Programm, das auf dem Heap Speicher für eine Zahl Typ int reserviert. Geben Sie die Adresse des reservierten Speichers mittels printf aus. Geben Sie den Speicher mittels free wieder frei. Danach geben Sie den Speicher wieder frei. Reservieren Sie erneut Speicher auf dem Heap und geben Sie die Adresse aus.