

## Programmering 1

### Laboration 7: Strukturer

Denna laboration är tänkt att både utföras och **redovisas** under de schemalagda timmar som finns anslagna för ändamålet. Ifall du inte har möjlighet att delta i laborationstillfällena, eller inte hinner slutföra laborationen under den schemalagda tiden måste du redovisa dina resultat i en **skriftlig laborationsrapport** som skickas in via Moodle. Ett obligatoriskt delkrav för att kunna bli godkänd i kursen är att **alla laborationer är godkända**.

**Obs: Vissa uppgifter (markerade med en asterisk) kommer att gås igenom gemensamt under laborationstillfällena. Dessa uppgifter godkänns alltså automatiskt om du deltar i laborationstillfället.**

#### 1. Enkel struct

**Att göra:** Skapa en **struct point3d\_t** som innehåller tre heltalskoordinater **x**, **y** och **z**.

Skriv därefter ett program som skapar två variabler av typ **point3d\_t**. Låt programmet initialisera en av variablerna med 'hårdkodade' värden, och låt användaren läsa in värden till den andra. Skriv därefter ut innehållet från både variablerna. En programkörning kan se ut enligt nedanstående:

```
Enter coordinates (x,y,z): 93 10 0
```

```
Hardcoded point: x=33, y=12, z=34
```

```
User-defined point: x=93, y=10, z=0
```

#### 2. Sammansatt struct

**Att göra:** Skapa en **struct box\_t** som innehåller en **struct point3d\_t** samt tre heltalsvärden **width**, **height**, **depth**.

Skapa en variabel av typ **struct box\_t**, låt användaren mata in värden och skriv ut de lagrade värdena. Beräkna även lådans volym och skriv ut denna.

```
Enter coordinates (x,y,z): 93 10 0
```

```
Enter dimension (w,h,d): 10 10 7
```

```
Properties of box:
```

```
x=93, y=10, z=0
```

```
width=10, height=10, depth=7
```

```
volume=700
```

### 3. *Räcka med structs*

Att göra: Skapa en räcka med utrymme för tre **struct box\_t**. Låt användaren mata in all information och skriv ut den inmatade informationen. Beräkna även lådornas totala volym.

```
Enter coordinates for box #1 (x,y,z): 93 10 0
Enter dimensions for box #1 (w,h,d): 10 10 7
Enter coordinates for box #2 (x,y,z): 0 0 0
Enter dimensions for box #2 (w,h,d): 1 1 1
Enter coordinates for box #3 (x,y,z): 10 10 10
Enter dimensions for box #3 (w,h,d): 5 5 5
```

You entered:

Box #1:

x=93, y=10, z=0

width=10, height=10, depth=7

Box #2:

x=0, y=0, z=0

width=1, height=1, depth=1

Box #3:

x=10, y=10, z=10

width=5, height=5, depth=5

Total volume: 826

### 4. *Typedef*

**Att göra:** Modifiera uppgift 3 så att du använder dig av **typedef** för att deklarerera strukturerna för punkter och lådor. Efter detta skall du alltså kunna definiera en variabel av typ **box\_t** genom att skriva

```
box_t box;
```

istället för

```
struct box_t box;
```

## 5. (\*) *Strukturer och funktioner*

**Att göra:** Skriv ett program som låter användaren mata in två klockslag och beräknar tidsskillnaden mellan dessa. Använd dig av en **struct time** som innehåller information om ett klockslag: timmar, minuter och sekunder. Om du vill kan du använda dig av **typedef**-alias.

Skapa en funktion **calculateTimeDiff()** som tar emot två **struct time** som parametrar och även returnerar resultatet som en **struct time**. Din **main**-funktion skall låta användaren mata in startvärdena och därefter anropa **calculateTimeDiff()**.

Programmet bör klara av klockslag **på olika sidor om midnatt**.

Programkörningarna kan t.ex. se ut som nedan.

```
Enter time #1 (HH:MM:SS): 14:00:25
Enter time #2 (HH:MM:SS): 15:15:30
Time difference is 1:15:05
```

```
Enter time #1 (HH:MM:SS): 23:00:00
Enter time #2 (HH:MM:SS): 1:15:15
Time difference is 2:15:15
```

Tips 1: `scanf("%d:%d:%d", ...);` // Inmatning med kolon som separator

Tips 2: `printf("%.02d", ...);` // Utskrift med minst två tecken, "leading zero"