



## Försättsblad till skriftlig tentamen vid Linköpings Universitet

<b>Datum för tentamen</b>	2013-04-30
<b>Sal (1)</b> Om tentan går i flera salar ska du bifoga ett försättsblad till varje sal och <u>ringa in</u> vilken sal som avses	TER3
<b>Tid</b>	8-10
<b>Kurskod</b>	TDDC74
<b>Provkod</b>	TEN1
<b>Kursnamn/benämning</b> <b>Provnamn/benämning</b>	Programmering - abstraktion och modellering Skriftlig tentamen/duggor
<b>Institution</b>	IDA
<b>Antal uppgifter som ingår i tentamen</b>	3
<b>Jour/Kursansvarig</b> Ange vem som besöker salen	Jalal Maleki
<b>Telefon under skrivtiden</b>	ankn. 19 63 eller 070-607 19 63
<b>Besöker salen ca kl.</b>	08:30 - 09:00
<b>Kursadministratör/kontaktperson</b> (namn + tfnr + mailaddress)	Anna Grabska Eklund, ankn. 2362, anna.grabska.eklund@liu.se
<b>Tillåtna hjälpmedel</b>	Inga
<b>Övrigt</b>	
<b>Vilken typ av papper ska användas, rutigt eller linjerat</b>	valfritt
<b>Antal exemplar i påsen</b>	

# Dugga

## **TDDC74 Programmering: Abstraktion och modellering Dugga 3 (provkod TEN1), Tid: kl 08-10, Datum: 2013-04-30**

Läs alla frågorna först och bestäm dig för den ordning som passar dig bäst.

Även om det i uppgiften står att du skall skriva en procedur/funktion, så får du gärna skriva ytterligare hjälpfunktioner som kan vara nödvändiga.

Det finns tre uppgifter i denna dugga. Poängen per uppgift/deluppgift anges i samband med varje uppgift.

Skriv tydligt för att öka läsbarheten. Använd **väl valda namn** på parametrar och **indentera** din kod. Sammanlagt två av poängen i Uppgift 3 är reserverade för väl-skriven kod, dvs, kod som innehåller bra namngivning, korrekt indentering, användning av dataabstraktion vid behov, och andra principer för bra Scheme-kod.

**Betygsgradering:** Det finns tre duggor i kursen. Varje dugga ger 12p, dvs totalt 36p för alla tre. För att passera en dugga krävs minst 3p på duggan. Totalt skall du på de tre duggorna för betyget 3 ha minst 18p, för betyget 4 minst 23p och för betyget 5 minst 27p.

Lycka till!

---

## Uppgift 1

(3p) Rita det omgivningsdiagram som avbildar evalueringen av följande tre uttryck. Numrera diagrammets delar så att det framgår i vilken ordning de skapas.

1. `(define *lamps* ())`

2. `(define (make-lamp)`

`(define status 'off)`

`(define (turn-me-off)`

`(set! status 'off))`

`(define (turn-me-on)`

`(set! status 'on))`

`(define (dispatch message)`

`(cond ((eq? message 'status) status)`

`((eq? message 'on)`

`(turn-me-on))`

`((eq? message 'off)`

`(turn-me-off))`

`(else (error "Invalid message: " message))))`

`(set! *lamps* (cons dispatch *lamps*))`

`dispatch)`

3. `(make-lamp)`

## Uppgift 2

(2p) Antag att vi vill lagra duggaresultaten för en student i en datastruktur med fyra delar: student-epost, dugga1, dugga2 och dugga3. T ex som

```
{asdfg333 4 6 . 3}
```

för studenten med epostadressen asdfg333 som fått skrivningspoängen 4, 6 och 3 för duggorna. Vi skulle kunna implementera följande ADT för detta ändamål.

```
(define (make-result student dugga1 dugga2 dugga3)
  (mcons student
    (mcons dugga1
      (mcons dugga2 dugga3))))

(define (get-email result)
  (mcar result))

(define (get-dugga1 result)
  (mcar (mcdr result)))

(define (get-dugga2 result)
  (mcar (mcdr (mcdr result))))

(define (get-dugga3 result)
  (mcdr (mcdr (mcdr result))))

(define (set-email! result student)
  ...)

(define (set-dugga1! result dugga1)
  ...)

(define (set-dugga2! result dugga2)
  ...)

(define (set-dugga3! result dugga3)
  ...)
```

Komplettera definitionerna på mutatorfunktionerna `set-email!`, `set-dugga1!`, `set-dugga2!`, och `set-dugga3!` som modifierar de olika delarna av resultatfälten till ett nytt värde. Vi kommer att använda oss av värdet -25 för de duggor för vilka inga resultat finns ännu. Exempel:

```
> (define exempel-resultat (make-result 'jalma123 10 12 -25))
> exempel-resultat
{jalma123 10 12 . -25}
> (set-dugga3! exempel-resultat 11)
> exempel-resultat
{jalma123 10 12 . 11}
```

## Uppgift 3

För att hålla reda på flera resultat av den typ som nämns i **Uppgift 2**, skall vi använda en global variabel `*TDDC74-results*` som är initialt tom.

```
(define *TDDC74-results* '())
```

Vi skriver en procedur `add-student` som lägger in en ny student i listan.

```
(define (add-student student-email)
  (let ((new (make-result student-email -25 -25 -25)))
    (set! *TDDC74-results* (cons new *TDDC74-results*))))
```

Observera att denna procedur, för enkelhetens skull, antar att studenten inte finns på listan. Vi använder denna procedur för att skapa fyra studenter i resultatlistan.

```
> (add-student 'abcef123)
> (add-student 'xyzab987)
> (add-student 'qwert345)
> (add-student 'asdfg333)
```

Värdet på `*TDDC74-results*` ser nu ut så här:

```
({asdfg333 -25 -25 . -25}
 {qwert345 -25 -25 . -25}
 {xyzab987 -25 -25 . -25}
 {abcef123 -25 -25 . -25})
```

### Deluppgift 3a)

(3.5p) Skriv proceduren `register-result` som tar tre argument, en epost, en duggabeteckning, och antal poäng (mellan 0 och 12) studenten fått på duggan och ändrar på lämpligt ställe i `*TDDC74-results*`. Tex,

```
(register-result 'abcef123 'dugga1 9)
(register-result 'asdfg333 'dugga1 4)
(register-result 'asdfg333 'dugga2 6)
(register-result 'asdfg333 'dugga3 3)
(register-result 'xyzab987 'dugga1 8)
(register-result 'xyzab987 'dugga2 11)
(register-result 'xyzab987 'dugga3 7)
```

ändrar listan till

```
({asdfg333 4 6 . 3}
 {qwert345 -25 -25 . -25}
 {xyzab987 8 11 . 7}
 {abcef123 9 -25 . -25})
```

### Deluppgift 3b)

(3.5p) Skriv proceduren `results-summary` som går genom `*TDDC74-results*` och returnerar en betyglista enligt exemplet nedan.

```
> (results-summary)
((asdfg333 . Underkänd)
 (qwert345 . Odefinierad)
 (xyzab987 . 4)
 (abcef123 . Odefinierad))
```

Betygen beräknas enligt följande tabell. För att klara av en enskild dugga behöver man minst 3 skrivningspoäng.

Betyg	Kommentar
Odefinierad	När summan av duggapoängen är ett negativt tal
Underkänd	När summan av duggapoängen är större än eller lika med 0 och dessutom gäller något av villkoren: 1) summan är mindre än 18 eller 2) poängen på någon dugga är mindre än 3
5	När poängsumman på de tre avklarade duggorna är större än eller lika med 27
4	När poängsumman på de tre avklarade duggorna är större än eller lika med 23 men mindre än 27
3	När poängsumman på de tre avklarade duggorna är större än eller lika med 18 men mindre än 23