

Försättsblad till skriftlig tentamen vid Linköpings Universitet

Datum för tentamen	2011-03-11
Sal (1) Om tentan går i flera salar ska du bifoga ett försättsblad till varje sal och <u>ringa in</u> vilken sal som avses	TER1
Tid	14-16
Kurskod	TDDC74
Provkod	TEN1
Kursnamn/benämning Provnamn/benämning	Programmering - abstraktion och modellering Skriftlig tentamen/duggor
Institution	IDA
Antal uppgifter som ingår i tentamen	4
Jour/Kursansvarig Ange vem som besöker salen	Anders Haraldsson
Telefon under skrivtiden	0705 147709, 281403
Besöker salen ca kl.	15
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Helene Meisinger, ankn. 1868, helene.meisinger@liu.se
Tillåtna hjälpmedel	Inga
Övrigt	
Vilken typ av papper ska användas, rutigt eller linjerat	
Antal exemplar i påsen	

AID-nummer:	Datum: 2011-03-11
Kurskod: TDDC74	Provkod: TEN1

1

Tekniska högskolan vid Linköpings universitet
Institutionen för datavetenskap
Anders Haraldsson

TDDC74 Programmering, abstraktion och modellering

DUGGA 3

fredag 11 mars 2011 kl 14-16

Uppgifterna löses direkt på denna uppgiftslapp, som lämnas in i sedvanligt tentamensomslag. Behöver du mer utrymme kan du lämna in extrablad.

Skriv tydligt så att inte dina lösningar missförstås. Använd väl valda namn på parametrar etc. Skriv lagom stort, en del skriver så smått så att vi har svårt att tolka texten och en del skriver så stort så att det inte ryms på raden. Börja raden så långt till vänster som möjligt.

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

Betygsgradering: Det är tre duggor. Varje dugga ger 12p, dvs. totalt 36p. För att passera en dugga krävs minst 3p på duggan. Totalt skall du på de tre duggorna för betyget 3 ha minst 20p. För betyget 4 minst 25p och för betyget 5 minst 30p.

Lycka till

2

AID-nummer:	Datum: 2011-03-11
Kurskod: TDDC74	Provkod: TEN1

Uppgift 1. Strukturändrande operationer på listor (3 poäng)

1a. (1p) Vi har skapat följande lista med muterbara cons-celler, som har bundits till variabeln `start`. Rita först upp vilka cons-cellerna med boxar och pilar (box-and-pointer diagrams) som skapats.

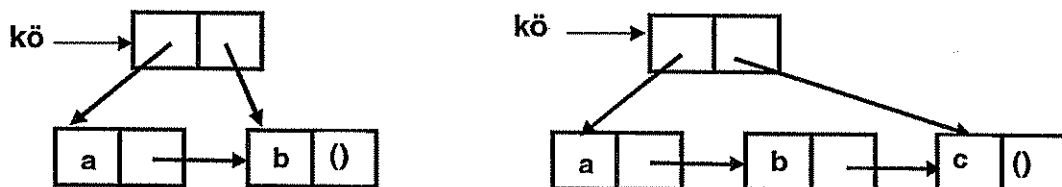
`start => {{a b} c d}`

Ange sedan vilka pekarändringar som utförs då nedanstående uttryck beräknas. Ange ändringarna i de cons-celler du angett ovan. Ange även vilket värde (i parentesformat) som skrivs ut av `start`.

```
(set-mcdr! (mcdr start) (mcar start))
start =>
```

```
(set-mcar! (mcar start) 'x)
start =>
```

1b. (2p) En kö representerades med ett köhuvud enligt följande modell:



Skriv först ett Scheme-uttryck, som skapar den *vänstra* strukturen och kallar värdet för **kö**.

Utgående från den vänstra strukturen, skriv sedan ett Scheme-uttryck, som lägger in elementet **c** sist i kön, genom att göra ändringar i den vänstra strukturen så att den *högra* strukturen erhålls. Du skall bara ge kod för detta specifika fall, ej en generell lösning.

AID-nummer:	Datum: 2011-03-11
Kurskod: TDDC74	Provkod: TEN1

3

Uppgift 2. Omgivningsdiagram (3 poäng)

2a. (1p) Använd exempel från nedanstående Scheme-exempel

```
(define c 1)
```

```
(define f  
  (lambda (x y)  
    (set! c (+ c 1))  
    (+ (* x y) 2)))
```

```
(f 3 c)
```

för att förklara följande:

När skapas en ny *ram*? Vart kopplas denna ram?

När skapas ett *procedurobjekt*?

När skapas nya *bindningar* i en ram?

Hur hittar man *värdet* på en variabel?

När kan ett värde i en bindning *ändras*?

4

AID-nummer:	Datum: 2011-03-11
Kurskod: TDDC74	Provkod: TEN1

Uppgift 2. Omgivningsdiagram, forts

2b. (2p) Vi har definierat Schemefunktionen f enligt följande:

```
(define (f a)
  (define (g b)
    (+ b a))
  (define (h a)
    (g (+ a 5)))
  (h (+ a 2)))
```

Vilket av följande värden är det korrekta?

(f 10) = 27

(f 10) = 29

Motivera ditt svar genom att rita upp omgivningsdiagrammet. För att kunna följa i vilken ordning de olika ramarna skapas, kallas de för E1, E2, E3 etc.

AID-nummer:	Datum: 2011-03-11
Kurskod: TDDC74	Provkod: TEN1

5

Uppgift 3. Procedurobjekt med tillstånd (2 poäng)

Följande uttryck ges till Scheme:

```
(define (värden värde-lista)
  (let ((lista värde-lista)
        (nästa-värde '()))
    (lambda (faktor)
      (set! nästa-värde (car lista))
      (set! lista (cdr lista))
      (if (null? lista) (set! lista värde-lista) (void))
      (* faktor nästa-värde))))
```

```
(define ett-till-två (värden '(1 2)))
```

```
ett-till-två =>
(ett-till-två 10) =>
(ett-till-två 20) =>
(ett-till-två 30) =>
```

Vilka värden returneras ovan?

Rita hur omgivningsdiagrammet ser ut efter det att ovanstående uttryck har beräknats. För att kunna följa i vilken ordning de olika ramarna skapas, kallas de för E1, E2, E3 etc. Ange vilka rammar och procedurobjekt som kan tas bort efter det att uttrycken har beräknats.

6

AID-nummer:	Datum: 2011-03-11
Kurskod: TDDC74	Provkod: TEN1

Uppgift 4. Objektorienterad modellering (4 poäng)

Nedan är ett klassiskt exempel på ett bankkonto, där man kan sätta in (deposit) resp. ta ut (withdraw) pengar. Dispatch-funktionen är ej fullständig.

```
(define (make-account balance)
  (define (withdraw amount)
    (if (>= balance amount)
        (begin (set! balance (- balance amount)) balance)
        "Insufficient funds"))
  (define (deposit amount)
    (set! balance (+ balance amount))
    balance)
  (define (dispatch .....
    (cond .....
      .....
      (else (error "Unknown request" m))))
    dispatch)
```

4a.(1p) Inom objektorientering har man ett antal begrepp: *klass*, *objekt*, *meddelande*, *konstruktör* och *metod*. Relatera dessa begrepp till ovanstående kod.

4b. (0,5p) Komplettera dispatch-funktionen om man vill kunna anropa på följande sätt:

```
(define acc-1 (make-account 100))
((acc-1 'deposit) 20)
((acc-1 'withdraw) 50)

(define (dispatch
```


AID-nummer:	Datum: 2011-03-11
Kurskod: TDDC74	Provkod: TEN1

7

Uppgift 4. Objektorienterad modellering, forts

4c. (0,5p) Komplettera dispatch-funktionen om man vill kunna anropa enligt följande:

```
(define acc-1 (make-account 100))  
(acc-1 'deposit 20)  
(acc-1 'withdraw 50)
```

```
(define (dispatch
```

4d. (2p) Antag att vi vill spara alla transaktionerna vid insättning resp. uttag. Vi önskar även kunna skriva ut alla transaktionerna gjorda för insättning resp. alla transaktionerna för uttag. Implementera detta i den objektorienterade modellen. Du får välja vilken av dispatch-alternativen du önskar.

Skriv koden nedan och markera var den skall läggas in.

