Försättsblad till skriftlig tentamen vid Linköpings Universitet

Datum för tentamen	*2010-12-20*
Sal	*SU17-18*
Tid	*14-18*
Kurskod	*TDP004*
Provkod	*DAT2*
Kursnamn/benämning	*Objekt-orienterad programmering*
Institution	*IDA*
Antal uppgifter som ingår i tentamen	*Teoretisk del: 5* *Praktisk del: 2*
Antal sidor på tentamen (inkl. försättsbladet)	*Teoretisk del: 4* *Praktisk del:7*
Jour/Kursansvarig	*Per-Magnus Olsson*
Telefon under skrivtid	*5607*
Besöker salen ca kl.	*Varje hel timme*
Kursadministratör (namn + tfnnr + mailadress)	*Anna Grabska-Eklund* *anna.grabska.eklund@liu.se*
Tillåtna hjälpmedel	*Teoretisk del: Inga* *Praktisk del: Den C++information som finns i systemet.*
Övrigt (exempel när resultat kan ses på webben, betygsgränser, visning, övriga salar tentan går i m.m.)	*Lösningsförslag kommer på kurshemsidan.*
Vilken typ av papper ska användas, rutigt eller linjerat	*Spelar ingen roll.*
Antal exemplar i påsen	*35*

Tentamen i TDP004
Objektorienterad Programmering
Praktisk del

Datum:	2010-12-20
Tid:	14-18
Plats:	SU-salar i B-huset.
Jour:	Per-Magnus Olsson, tel 285607 Jourhavande kommer att besöka skrivsalarna ungefär varje timme under skrivtiden.
Hjälpmedel:	Teoretisk del: Inga. Praktisk del: Den C++ information som finns i systemet.
Betygsättning:	Max antal poäng: 40 med 20 poäng vardera på teori och praktikdel.

Poäng	**Betyg**
35-40	**5**
28-34	**4**
21-27	**3**
0-20	**U**

Anvisningar:	Börja med den teoretiska delen. När du är klar med den lämnar du in den och får den praktiska delen. När du har lämnat in den teoretiska delen kan du inte återvända till den. Uppgifterna är inte ordnade i svårighetsgrad. En kort hjälptext till Eclipse finns på sid 2-3.

Lycka till!

Kort Eclipsehjälp

Om du inte är bekant med Eclipse rekommenderas att du läser igenom följande guide.

Att komma igång
Skriv eclipse & i terminal. Efter viss väntetid kommer ett fönster att visas. Välj ”C++ Perspective” i övre högra hörnet av fönstret. ”C++ Perspective” finns under ”Other”.

Välj ”Managed Make C++ Project” som din typ av projekt. Gör ett projekt för varje uppgift, om inte annat sägs i uppgiften.

Ovanstående är viktigt för att Eclipse ska bete sig korrekt med avseende på kompilering etc.

Inkludera eventuella givna filer
För att använda programfiler som finns i katalogen given_files, kopiera dem till ditt workspace. Högerklicka sedan på ”C/C++ Project” vilken normalt är en vertikalt fönster finns längst till vänster i huvudfönstret. Välj sedan ”Refresh” i den meny som visas. De inkluderade filerna visas nu tillsammans med dina egna filer. Programfilerna kan nu inkluderas som vanligt.
Datafiler behöver endast kopieras till ditt workspace.

Kompilera
I ”C/C++ Project”, högerklicka på projektet och välj ”Build” i menyn.

Köra program
I ”C/C++ Project”, välj ”Choose binaries” och välj din binär, vilket normalt har samma namn som filen där main är definierad. Högerklicka och välj ”Run as local C++ program”. Det går även att välja ”Run”-knappen i menyn längst upp i fönstret.

Automatisk kompilering
Som default är Eclipse inställt på att kompilera alla filer så snart du har slutat skriva. Om du vill avaktivera den funktionen, gå till ”Project”-menyn och vid texten ”Build Automatically”, ta bort krysset i rutan.

Ta bort alla kompilerade filer
Om du vill ta bort alla kompilerade filer (källkodsfiler är opåverkade): gå till ”Project”-menyn och välj ”Clean”. Då få du upp en ruta där du väljer lämpligt alternativ. Observera att det här finns en ruta vid texten ”Build

Automatically” vilket är ikryssad och kommer att kompilera dina filer (även om du har tahit bort krysset vid ”Build Automatically” enligt ovanstående punkt.

Om fönster inte visas
Om konsollen med resultatet av programkörningar inte visas, så kan du aktivera visningen av fönstret genom att gå att ”Project”-menyn, och välja ”Show vindow” och sedan välja ”Console”. Om ”Problems”-fönstret som visar eventuella komplieringsfel inte visas, gör samma sak men välj ”Problems” istället.

Om Eclipse verkar ha låst sig
Prova att växla mellan olika fönster med Alt-Tab. Ibland hamnar pop-up-fönster som väntar på din input bakom huvudfönstret. Om det inte fungerar kanske du måste stänga Eclipse från konsollen och starta om.

TDP004 Objektorienterad Programmering
Praktisk del

1. På julbordet finns en mängd olika rätter, och en del av dem måste hållas vid en viss varm temperatur, medan andra kan serveras vid vilken temperatur som helst. Eftersom det går åt el för att hålla värmen på maten och elen kostar pengar, vill en kostnadsmedveten krögare beräkna ungefär hur mycket energi per timme det kommer att gå åt för att hålla maten vid rätt temperatur.

   För enkelhetens skull antar vi att det bara finns tre olika maträtter: skinka, potatis och köttbullar. Skinkan behöver inte hållas varm (d.v.s. det går inte åt någon energi alls för den), medan potatisen och köttbullarna måste hållas varma. Potatisen ska hållas vid temperaturen 50 grader och köttbullarna ska hållas vid temperaturen 70 grader.

   Eftersom potatis är mindre populärt än köttbullar så är åtgången mindre och man kan förenkla beräkningen genom att anta att det går åt lika mycket effekt att hålla potatisen uppvärmd oavsett mängd, och i det här fallet går det åt 6,5W.

   För köttbullarna krävs det däremot effekten 10,0 W per kilo, för att hålla dem på 70 grader. Eftersom köttbullarna äts up och fylls på under dagen så kan man räkna med att det i genomsnitt finns fyra kilo köttbullar som måste värmas upp.

   Din uppgift är att implementera en arvshierarki med subklasser för de tre olika maträtterna. I basklassen ska det finnas en virtuell funktion som beräknar och returnerar den mängd energi mätt i Wh (se nedan) som går åt för att hålla en viss maträtt på korrekt temperatur i en timme. Där det är lämpligt ska funktionen överlagras i subklasserna. (10p)

   Observera att:
   - energi = effekt*tid, där tiden mäts i sekunder.
   - Energi mäts i Joule (J) och effekt mäts i Watt (W). 1 J = 1 Ws (wattsekund)
   - 1 Wh = 3600 J = 3600 Ws.

2. Ett sätt att undvika att någon annan kan läsa din önskelista till tomten är att chiffrera den. Därför ska du göra ett chiffer av typen Groensfeldchiffer. Ett sådant chiffer fungerar så att man först anger en chiffreringsnyckel och sedan anger texten som ska chiffreras. Med hjälp av nyckeln utförs sedan chiffrering och dechiffrering enligt nedan.

   Användaren ska kunna mata in en nyckel och sedan texten som ska chiffreras. Man avbryter programmet med Ctrl-C. När nyckeln och texten har angetts ska det chiffrerade meddelandet skrivas ut, och sedan ska detta dechiffreras igen, och det dechiffrerade meddelandet skrivas ut. Se nedanstående exempel (användarens input understruken):

   ```
   Ange nyckel: 5273
   Ange meddelande: HEMLIGT MEDDELANDE
   Chiffrerat meddelande: MGTONIAPJFKHQCUGJ
   Dechiffrerat meddelande: HEMLIGTMEDDELANDE
   ```

   För att chiffrera ett meddelande använder man en chiffreringsnyckel som består av siffror 0-9. Både avsändaren och mottagare måste förstås använda samma nyckel. Antag att nyckeln är `5273` och det texten som ska chiffreras är ”`HEMLIGT MEDDELANDE`”. Man upprepar sedan chiffreringsnyckeln under bokstäverna enligt följande:

   ```
   HEMLIGT MEDDELANDE
   5273527 3527352735
   ```

Vid chiffreringen går man igenom meddelandet bokstav för bokstav och ersätter varje bokstav med den bokstav som kommer det antal positioner *efter* i alfabetet som den motsvarande siffran i nyckel anger. H+5 ger M, E+2 ger G, M+7 ger T, L+3 ger O, osv. Efter att alla bokstäver i meddelandet har gåtts igenom erhålls det chiffrerade meddelandet som:

```
MGTONIAPJFKHQCUGJ
```

Alfabetet hanteras cirkulärt så att alla bokstäver kan chiffreras. Om man t.ex. ska chiffrera bokstaven V (femma från slutet i alfabetet) och siffran i kodnyckeln är 7 så ska man välja bokstaven C (7 steg framåt från V: **V**WXYZAB**C**…)

Dechiffrering av ett meddelande görs omvänt, man ersätter varje bokstav med den bokstav som kommer det antal steg *före* i alfabetet som den motsvarande siffran i nyckeln anger. M-5 ger H, G-2 ger E, T-7 ger M, O-3 ger L, osv. När alla bokstäver dechiffrerats erhålls (nästan) det ursprungliga meddelandet :

```
HEMLIGTMEDDELANDE
```

Som synes ovan får mellanslag ignoreras. Du kan anta att meddelandet enbart innehåller stora bokstäver och inte Å, Ä, Ö. Dvs, alfabetet för meddelandena är:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

Tips 1! Man kan få fram ASCII-koden för ett tecken genom:

```
char A = 'A';
int ascii_for_a = static_cast<int>(a);        //ascii_for_a = 65
```

Tips 2!
Använd

```
fflush(stdin);
```

efter att du har läst nyckeln för att undvika eventuella problem med inläsningen av meddelandet.

Sista bladet på tentamen innehåller en ASCII-tabell.

Gör en klass `GroensfeldCipher` med funktionerna `Cipher` och `Decipher`. Båda tar som inparameter ett meddelande och en nyckel. Funktionerna ska returnera den chiffrerade texten respektive den dechiffrerade texten. Implementera eventuella hjälpfunktioner på lämpligt sätt. (10p)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL