

# Tentamen i TDP004

## Objektorienterad Programmering

### Praktisk del

- Datum: 2011-04-28
- Tid: 08-12
- Plats: SU-salar i B-huset.
- Jour: Per-Magnus Olsson, tel 281456  
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: 42 med 21 poäng vardera på teori och praktikdel.
- | Poäng | Betyg    |
|-------|----------|
| 36-42 | <b>5</b> |
| 29-35 | <b>4</b> |
| 22-28 | <b>3</b> |
| 0-21  | <b>U</b> |

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.

Skriv svaret på varje teoretisk uppgift på ett separat blad.

Uppgifterna är inte ordnade efter svårighetsgrad.

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.

Du måste även skapa en fil som har funktionen `main` definierad. Det går bra att skapa en `.cpp`-fil och där definiera funktionen `main` enligt nedan, vilken blir det som operativsystemet anropar du när exekverar programmet.

```
int main()
{
//Övriga anrop görs här.
return 0;
}
```

Om du inte skapar en `main`-funktion så kommer kompilatorn att säga att funktionen `main` saknas och att ett allvarligt symbolreferensfel har inträffat.

**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 ett vertikalt fönster finns längst till vänster i huvudfönstret. Välj sedan "Refresh" i den meny som visas. De inkluderade 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. Som default är Eclipse inställt på att automatiskt kompilera under tiden du programmerar.

## Om kompileringen krånglar

Högerklicka på projektet och välj "Properties" -> C/C++ Build. Välj fliken "Build Settings".

Klicka bort "Use default command". I rutan under, skriv "gmake -k". Klicka på OK i nedre högra hörnet.

## 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 klicka på texten "Build Automatically".

### **Ta bort alla kompillerade filer**

Om du vill ta bort alla kompillerade filer (källkodsfiler är opåverkade): gå till "Project"-menyn och välj "Clean".

### **Köra program**

I "C/C++ Project", välj "Run as"->"Run as local C/C++ application".

Det går även att välja "Run"-knappen i menyn längst upp i fönstret.

### **Om fönster inte visas**

Om konsollen med resultatet av programkörningar inte visas, så kan du aktivera visningen av fönstret genom att gå till "Project"-menyn, och välja "Show window" och sedan välja "Console".

Om "Problems"-fönstret som visar eventuella kompileringsfel 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

### OBS!

Det finns vissa problem med den version av Eclipse som används i tentasystemet. Bland annat kan det vara krångel med att Eclipses defaultinställningar för att kompilera inte fungerar och man får ett felmeddelande liknande:

```
make -k all
```

```
make: Fatal error in reader: makefile, line 6: Unexpected end of line seen
```

Då bör man ändra så att `gmake` används istället för `make`, vilket man gör på följande sätt: högerklicka på ditt projekt och välj "Properties" och sedan "C/C++ Build". Välj fliken "Build Settings". Klicka bort "Use default command". I rutan under, skriv `gmake -k`. Klicka på OK i nedre högra hörnet.

Det kan också finnas problem med att köra sitt program från Eclipse. Då får man istället köra sitt program från ett terminalfönster. Om den exekverbara filen heter `a.out` exekveras denna med kommandot `./a.out`

### OBS!

- 1) Inom matematiken finns något som kallas Maclaurinserier, döpt efter matematikern Colin Maclaurin. Med hjälp av en Maclaurinserie kan man approximera (uppskatta) olika funktioner runt  $x=0$ . För att uppskatta exponentialfunktionen  $e^x$  väldigt noggrant kan följande Maclaurinserie användas:  $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$ . Observera att antalet termer i serien är oändligt. Man kan också göra en lite mindre noggrann approximation genom att begränsa antalet termer, men samma formel för varje term används fortfarande. Om man vill approximera  $e^x$  med 3 termer så blir resultatet  $1 + x + \frac{x^2}{2}$  och om man vill approximera med 6 termer så blir resultatet

$$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120}.$$

Skriv en funktion som gör approximationer av  $e^x$  med hjälp av Maclaurinserien enligt ovan. Din funktion ska ta antalet termer som inparameter, beräkna det generella uttrycket enligt ovan och skriva ut det totala uttrycket med alla termerna på skärmen. Det ställs inget krav att texten/termerna ska formatteras som ovan, men det måste klart framgå vad som hör till vilken term. För full poäng krävs en effektiv implementation. (6p)

Tips! Nollfakultet ( $0!$ ) är lika med 1. Givet formeln och exemplen ovan bör du kunna lista ut hur serien ska se ut med olika termer.

- 2) Du ska implementera ett system för att hålla ordning på böckerna i ett bibliotek. Följande information ska sparas för varje bok: titel, författarens namn, antal sidor i boken, utgivningsår, förlag samt ISBN-nummer. Ett ISBN-nummer är ett unikt nummer för varje bok (jämför med personnummer).
- a) Din uppgift är att skapa en lämplig datastruktur för att kunna spara ovanstående information om böckerna. Varje instans av datastrukturen ska spara information om en bok. (3p)
- b) Antag att du har skapat din datastruktur enligt a-delen av uppgiften. Du ska nu välja en lämplig STL-container för att spara en stor mängd instanser som representerar böcker. Böckerna ska läggas in i din container när registret skapas och de tas sällan bort från systemet. Det som sker oftast är att en låntagare söker efter en bok med hjälp av ISBN-numret.
- Bestäm vilken STL-container är mest lämplig för registret. Skapa två instanser av din datastruktur som representerar böcker, från a)-delen av uppgiften, och lägg till dem i registret. I samma fil, skriv en kort text som motiverar ditt val av container. (3p)
- c) Skriv en funktion som låter användaren söka efter en bok med hjälp av ISBN-numret. Funktionen ska som inparameter ta ISBN-numret och vad övrigt du tycker är lämpligt. Returvärdet är en pekare till boken. Kom ihåg att hantera fallet med att ingen bok med ISBN-numret hittades. (3p)
- d) Även om det sällan händer att en bok tas bort så händer det till exempel när en slarvig låntagare tappar bort en bok. Skriv en funktion som tar bort en bok från den container du har valt. Inparameter till funktionen är ISBN-numret. (3p)
- e) Biblioteket ska nu börja tillhandahålla CD-skivor för utlåning. För varje CD ska följande information sparas: artist, namnet på skivan, skivbolag som get ut skivan, utgivningsår, samt namn och längd (i minuter och sekunder) på alla låtarna på skivan. Vilka ändringar kräver detta i den datastruktur som du skapade i del a)? Finns det något som går att återanvända? (3p)