

EXAM PROBLEMS at the University of Oslo must be typeset according to very rigid specifications; two examples are shown in Figures 1 and 2 on pages 2 and 3. The L^AT_EX document class `uioexam` has been written to implement these rules. It is based on the standard `article` style with the following main modifications:

- The top half of the front page contains a table with the most important information regarding the exam.
- The page header gives the exam subject and the date, and the footer contains a reference to the following page to make sure that no pages are missing.

Program documentation system

This program was documented using the `web0` package which is based on Donald Knuth's ideas of *literate programming*. For more information on the `web0` implementation, see <http://dag.at.ifi.uio.no/public/doc/web0.pdf>.

1 User guide for the `uioexam` document class

The `uioexam` document class is used for typesetting exam problems at the University of Oslo. Two simple examples are shown in Figures 1 and 2 on the following pages. (Both examples are two pages long to demonstrate the special page headers and footers.)

1.1 Document class parameters

The `uioexam` class accepts these parameters:

11pt selects 11 pt type size.¹

12pt chooses 12 pt type size; this is the default.

american or **USenglish** is used when the exam text is written in American English; this is the default.

del is used for constituent² exams which will be marked as “Constituent exam” or “Deleksamen” (depending on the language). This option may be used in combination with the **ny** and **utsatt** options.

english or **UKenglish** is used for British English.

exercise will call the individual questions (created using the `\oppgave` command) for “Exercises” (see Section 2.3.2.1 on page 9). (This option should only be used for English text.)

norsk is for exams written in Norwegian “Bokmål”.

number will print a small box for the candidate's number; see an example in the top right-hand corner in Figure 2 on page 3.

ny is for new³ exams, and it may be used in combination with the **del** and **utsatt** options.

nynorsk is for writing exams in Norwegian “Nynorsk”.

¹10 pt type size is not allowed; it is too small for this kind of document.

²A *constituent exam* is an exam which is taken during the term and contributes a certain percentage to the final grade.

³A *new exam* may be taken if you fail the ordinary exam.

UNIVERSITETET I OSLO
Det matematisk-naturvitenskapelige fakultet

Eksamen i: INF2100 – Programmeringslaboratorium
med kompilatorkonstruksjon
Eksamensdag: 17. oktober 2008
Tid for eksamen: 9.00–12.00
Oppgavesettet er på 2 sider.
Vedlegg: Ingen
Tillatte hjelpemidler: Alle

Kontroller at oppgavesettet er komplett før
du begynner å besvare spørsmålene.

Innhold

1 Oversettelse (vekt 40%)	side 1
2 Programmering (vekt 50%)	side 2
3 Hva kan du ellers? (vekt 10%)	side 2

Oppgave 1 Oversettelse (vekt 40%)

Her er et C-program. Oversett det til Java. (Dette er det første av
3 problemer på 2 sider.)

```
/* Program 'gcd'
-----
A function to compute the greatest common divisor.
*/
int LF; /* Line feed */

int gcd (int a, int b)
{ /* Computes the gcd of a and b. */

    while (a != b) {
        if (a < b) {
            b = b-a;
        } else {
            a = a-b;
        }
    }
    return a;
}
```

(Fortsettes på side 2.)

Eksamen i INF2100, 17. oktober 2008

Side 2

```
int main ()
{
    int v1, v2, res;

    v1 = getint(); v2 = getint();
    res = gcd(v1,v2);
    LF = 10;
    putint(res); putchar(LF);
}
```

Oppgave 2 Programmering (vekt 50%)

Skriv parse-metoden til ifUnit.

2a Forklaring (vekt 10%)

Vis hvorledes synkroniseringen med skanneren skjer.

Oppgave 3 Hva kan du ellers? (vekt 10%)

Skriv om noe du tror du kan.

```
1 \documentclass[norsk]{uioexam}
2 \usepackage[utf8]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage{babel,textcomp,fancyvrb}
5 \usepackage{newcent} %% Lettere å lese enn «Computer Modern»
6
7 \dato{17.~oktober 2008}
8 \emne{INF2100}{Programmeringslaboratorium\
9     med kompilatorkonstruksjon}
10 \tid{9.00}{12.00}
11
12 \begin{document}
13 \tableofcontents
14
15 \oppgave[40\%]{Oversettelse}
16 Her er et C-program. Oversett det til Java. (Dette er det
17 første av \Nproblems~problemer på \Npages~sider.)
18 \VerbatimInput[fontsize=\small]{gcd.rusc}
19
20 \oppgave[50\%]{Programmering}
21 Skriv \texttt{parse}-metoden til \texttt{ifUnit}.
22
23 \deloppgave[10\%]{Forklaring}
24 Vis hvorledes synkroniseringen med skanneren skjer.
25
26 \oppgave[10\%]{Hva kan du ellers?}
27 Skriv om noe du tror du kan.
28 \end{document}
```

Figure 1: Norwegian exam demo

UNIVERSITY OF OSLO

Faculty of mathematics and natural sciences

Candidate no

Deferred constituent exam in: INF2270 – Computer Architecture
Day of examination: 17th October 2008
Examination hours: 14:30 – 17:30
This problem set consists of 2 pages.
Appendices: The x86 instruction set table
The gas assembly manual
Permitted aids: A calculator
Any written or printed material

Please make sure that your copy of the problem set is complete before you attempt to answer anything.

Contents

1 Translation (weight 75%) page 1
2 Multiple choice (weight 25%) page 2

Problem 1 Translation (weight 75%)

Translate this C program into x86 assembly language:

```
#include <stdio.h>

void rle (char *to, char *from)
{
    char *t = to, *f = from;

    while (*f) {
        char c = *(f++);
        int n = 1;

        while (n < 9 && *f == c) {
            ++n; ++f;
        }
        if (n == 1) {
            *(t++) = c;
        } else {
            *(t++) = '#'; *(t++) = '0'+n; *(t++) = c;
        }
    }
    *t = 0;
}

(Continued on page 2.)
```

Exam in INF2270, 17th October 2008 Page 2

```
int main (void)
{
    char s[200];

    rle(s, "Abba spiller musikk");
    printf("Svaret er: %s.\n", s);
    return 0;
}
```

Problem 2 Multiple choice (weight 25%)

Which of these are legal instruction names in x86 assembly code?

AddL
 ModB
 JumpEqual

```

1 \documentclass[UKenglish,number,utsatt,del]{uioexam}
2 \usepackage[utf8]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage{babel,textcomp,fancyvrb}
5 \usepackage{mathpazo} %% Easier to read than "Computer Modern"
6
7 \date{17th October 2008}
8 \emne{INF2270}{Computer Architecture}
9 \tid{14:30}{17:30}
10 \hjelpemidler{A calculator\\ Any written or printed material}
11 \vedlegg{The x86 instruction set table\\
12 The \texttt{gas} assembly manual}
13
14 \begin{document}
15 \tableofcontents
16
17 \oppgave[75\%]{Translation}
18 Translate this C program into x86 assembly language:
19 \VerbatimInput[fontsize=\small]{rle.c}
20
21 \oppgave[25\%]{Multiple choice}
22 Which of these are legal instruction names in x86 assembly code?
23 \begin{choicelist}[]
24 \choice AddL \choice ModB \choice JumpEqual
25 \end{choicelist}
26 \end{document}

```

Figure 2: English exam demo

plainsub Normally, subproblems (started by a `\deloppgave` command) will be numbered “1a”, “1b”, etc. Using the `plainsub` option will result in just “a”, “b”, etc.

problem will call the individual questions (created using the `\oppgave` command; see Section 2.3.2.1 on page 9) for “Problems”; this is the default. (This option should only be used for English text.)

utsatt is used for deferred⁴ exams. This option may be used in combination with the **del** option already mentioned.

1.2 Exam information

Formal information about the exam is supplied by special commands,⁵ usually placed just before `\begin{document}`.

`\dato{date}` provides the exam date.

`\emne{code}{name}` specifies the course; the first parameter gives the code (as in “INF1000”) and the second one the full name.

`\hjelpemidler{text}` is used to provide information on which aids are allowed during the exam; the default is “Any”.

`\tid{start time}{end time}` give the time for the exam.

`\vedlegg{text}` tells which appendices are supplied with the exam text; the default is “None”.

If a *name* or a *text* is too long, you may add `\\s` to split the lines.

1.3 The problems

Each new problem should be started with an `\oppgave{...}`. The parameter should be a suitable problem title.

Exam writers are requested by the faculty to assign weights to the individual problems; this is easily accomplished using an option to `\oppgave`, as in

```
\oppgave[20\%]{Translation}
```

1.3.1 Subproblems

A problem may be split into several parts using the `\deloppgave` command. Parameter and option are as for `\oppgave`.

1.4 Useful declarations

1.4.1 List of problems

The command `\tableofcontents` will print a list of all the problems. This is requested by the faculty.

1.4.2 How many problems are there?

The command `\Nproblems` will tell how many problems there are in the set.

⁴A *deferred exam* is an exam you take when you were ill during the standard exam.

⁵The names of these commands reveal the Norwegian origin in this document class.

1.4.3 How many pages in the problem set?

The command `\Npages` returns the number of pages in the problem set.⁶

1.4.4 Multiple choice lists

The environment `\begin{choicelist}... \end{choicelist}` is used for a list of alternatives from which the candidate shall make his or her selection. Each alternative is indicated by a `\choice`;⁷ for an example, see line 24 in Figure 2 on page 3.

Normally, the choices are labeled “a”, “b”, etc. If you want a different label (or no labels at all), just add an option to the `\begin{choicelist}... \end{choicelist}` environment (as has been done in Figure 2; see line 23).

⁶The result from `\Npages` may be wrong if there is floating material (`\begin{figure}... \end{figure}` or `\begin{table}... \end{table}`) at the end of the document; try to avoid this.

⁷For historical reasons, the command `\item` may also be used; it has exactly the same effect as `\choice` in this context.