Tentamen i TDP004
Objektorienterad Programmering
Teoretisk del

Datum:	2008-12-16
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: 44 med 22 poäng vardera på teori och praktikdel.

Poäng	**Betyg**
39-46	**5**
31-38	**4**
24-30	**3**
0-23	**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. Skriv svaret på varje teoretisk uppgift på ett separat blad. Uppgifterna är inte ordnade efter svårighetsgrad.

Lycka till!

TDP004 Objektorienterad Programmering
Teoretisk del

1. Du har fått två versioner av destruktorn i en klass, enligt nedan. Vilken av dessa två versioner bör vara snabbast? Varför? (2p)

objectvector är en medlemsvariabel av typen std::vector som innehåller en mycket stor mängd element av typen LargeObject*.

Version 1.
```
ExampleClass::~ExampleClass()
{
    std::vector<LargeObject*>::iterator i;
    for(i = objectvector.begin(); i != objectvector.end(); i++)
    {
        delete (*i);
    }

    // I övrigt samma som den andra versionen.
}
```

Version 2.
```
ExampleClass::~ExampleClass()
{
    std::vector<LargeObject*>::reverse_iterator i;
    for(i = objectvector.rbegin(); i != objectvector.rend(); i++)
    {
        delete (*i);
    }
    // I övrigt samma som den första versionen.
}
```

2. I en header-fil finns ett interface för en viss datastruktur som du är intresserad av, enligt nedan. Du har inte tillgång till vare sig cpp-filen för klassen Stack eller någon mera information om klassen StackImplementation. Vilka objektorienterade principer kan detta sätt att implementera en funktion sägas vara exempel på? Motivera ditt svar. (4p)

```
class StackImplementation;

class Stack
{
public:
    int pop();
    int peek();
    void push(int value);

private:
    StackImplementation* impl;
};
```

3. Funktionerna f och g enligt nedan. Vad kommer att skrivas ut på skärmen i funktionen g? Motivera ditt svar. (6p)

```
void f(int c, int& d, double* e)
{
    ++c;
    d /= c;
    e  = 0;
}

void g()
{
    double x     = 12.5;
    int y        = 2;
    int z        = 3;

    f(z, y, &x);

    std::cout << x << std::endl;
    std::cout << y << std::endl;
    std::cout << z << std::endl;
}
```

4a) Vad behöver du ändra i nedanstående kod om du vill byta container från std::vector till std::list? (7p)

```
double find_greatest(const std::vector<double>& data) const
{
        double greatest = 0.0;
        unsigned int i = 0;
        bool firstTime = true;
        for(; i < data.size(); i++)
        {
                if ( (true == firstTime) || (data[i] > greatest))
                {
                        firstTime = false;
                        greatest = data[i];
                }
        }
        return greatest;
}
```

4b) Vad är poängen med att argumentet till funktionen är av typen const std::vector<double>&? Vad kallas det sättet att överföra data? Vad kallas det alternativa sättet? Vad skulle funktionens argument vara i det fallet? (4p)