CHALMERS LINDHOMEN
Institutionen för data- och elektroteknik
Uno Holmer
Uno.Holmer@chl.chalmers.se

Lösningsförslag till tentamen* *Preliminär*

Kursnamn	**Objektorienterad programutveckling**
Tentamensdatum	**2000-08-14**
Program	**DAI 2**
Läsår	**99/00, lp III/IV**
Examinator	**Uno Holmer**

* se även `http://www.chl.chalmers.se/~holmer/` där finns det mesta av kursmaterialet.

Uppgift 1 (12 p)

Ingen lösning ges. Se kurslitteraturen.

Uppgift 2 (10 p)

```
class Minnestarning : public Tarning {
public:
    Minnestarning() : Tarning() {}
    void Kasta();    // definiera om Tarning::Kasta()
};

void Minnestarning::Kasta() {
    int Gamla = Uppsida();
    do
        Tarning::Kasta();
    while ( Uppsida() == Gamla );
}
```

Uppgift 3 (14 p)

a) (7 p)

```
class ExtendedAnimation : public BasicAnimation{
public:
    ExtendedAnimation() : Angle(0) {}
    void Rotate( int Turns );
protected:
    double Angle;
};

void ExtendedAnimation::Rotate( int Turns ) {
    int SavedAngle = Angle;
    for ( Angle = 0; Angle < Turns*360; Angle++ ) {
        Draw();
        delay( StepRate );
        Erase();
    }
    Draw();
    Angle = SavedAngle;
}
```

b) (7 p)

```
class Rectangle : public ExtendedAnimation {
public:
    Rectangle( int W, int H ) : Width(W), Height(H) {}
    void Draw();
    void Erase();
private:
    int Width, Height;
    void Draw( int Color );
};

void Rectangle::Draw( int Color ) {
    setcolor( Color );
    int Mid = X + Width/2, W = Width/2*Cos( Angle );
    rectangle( Mid - W, Y, Mid + W, Y + Height );
}

void Rectangle::Draw() { Draw( RED ); }

void Rectangle::Erase() { Draw( BLACK ); }
```

Uppgift 4 (10 p)

Fältet Table innehåller basklasspekare som i loopen initieras till att peka på slumpmässiga objekt av subklasserna S1-S3. Samtliga pekare har statisk typ *B, men varierande dynamiska typer S1-S3, beroende på vad de pekar på. I den andra loopen anropas G för alla objekten. G är ej virtuell varför pekarnas statiska typ *B gör att B::G alltid väljs – även om ett S2-objekt utpekas. Om ett S1- eller S3-objekt pekas ut anropar G resp. subklass omdefinition av F, men för ett S2-objekt anropas B::F eftersom S2 ej definierar om F.

a) 135 = 1*3*3*3*5. Detta resultat fås om fältets pekare pekar ut ett objekt av klass S1, tre S3, samt ett S2, i godtycklig ordning.
b) 450: Omöjligt. Enda möjligheten att erhålla en jämn produkt är att funktionen G i S2 anropas vilket ej kan ske enligt ovanstående resonemang.
c) 35: Omöjligt, talet innehåller primtalet 7 som faktor vilket inte returneras av någon av funktionerna.

Uppgift 5 (12 p)

a) (6 p)

```
class Cell : public Storable {
public:
    Cell( const char *aFileName );
    void Set( int aValue );
    int Get();
    void Save();
    bool Load();
private:
    int theValue;
    bool Defined;
};

Cell::Cell( const char *aFileName ) : Storable(aFileName) {
    Defined = false;
}
```

```
void Cell::Set( int aValue ) {
    theValue = aValue;
    Defined = true;
}

int Cell::Get() {
    if ( Defined )
        return theValue;
    else
        throw "Cell: value is undefined";
}

void Cell::Save() {
    ofstream Out( theFileName );
    if ( ! Out )
        throw "Cell: Cannot create file";
    if ( Defined )
        Out << theValue;
    Out.close();
}

bool Cell::Load() {
    ifstream In( theFileName );
    if ( ! In )
        throw "Cell: cannot open file";
    int Temp;
    In >> Temp;
    if ( ! In.eof() ) {
        theValue = Temp;
        Defined = true;
        return true;
    }
    In.close();
    return false;
}
```

b) (5 p)

```
class ObjectManager {
public:
    void Register( Storable &Obj );
    void LoadAll();
    void SaveAll();
private:
    List<Storable*> theObjects;
};

void ObjectManager::Register( Storable &Obj ) {
    ListItr<Storable*> Itr(theObjects);
    Itr.Insert( &Obj );
}

void ObjectManager::LoadAll() {
    ListItr<Storable*> Itr(theObjects);
    for ( Itr.First(); +Itr; ++Itr )
        Itr()->Load();
}
```

```
void ObjectManager::SaveAll() {
    ListItr<Storable*> Itr(theObjects);
    for ( Itr.First(); +Itr; ++Itr )
        Itr()->Save();
}
```

c) (3 p)

```
void main() {
    Cell Allan( "Allan.txt" ), Bellan( "Bellan.txt" ),
         Cellan( "Cellan.txt" );

    ObjectManager M;
    M.Register( Allan );
    M.Register( Bellan );
    M.Register( Cellan );
    M.LoadAll();

    int I, J , K;
    cout << "Skriv tre tal: ";
    cin >> I >> J >> K;
    Allan.Set( Allan.Get() + I );
    Bellan.Set( Bellan.Get() + J );
    Cellan.Set( Cellan.Get() + K );

    M.SaveAll();
}
```