

# Tentamen i TDP004

## Objektorienterad Programmering

### Teoretisk del

- Datum: 2008-08-14
- Tid: 08-12
- Plats: PC6-PC7 i E-huset.
- Jour: Per-Magnus Olsson, tel 285607
- Jourhavande kommer att besöka skrivsalarna varje timme under skrivtiden.
- Hjälpmedel: Teoretisk del: Inga.  
Praktisk del: En bok om C++.
- Betygsättning: Max antal poäng: 44 med 22 poäng vardera på teori och praktikdel.
- | Poäng | Betyg    |
|-------|----------|
| 38-44 | <b>5</b> |
| 30-37 | <b>4</b> |
| 23-29 | <b>3</b> |
| 0-22  | <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 uppgift på ett separat blad.
- Uppgifterna är inte ordnade efter svårighetsgrad.

Lycka till!

# TDP004 Objektorienterad Programmering

## Teoretisk del

1. En viss funktion finns i tre versioner: A, B och C enligt nedan. Det enda som skiljer dem åt är inparametern.

- A. `double Calculate(LargeObject object)`
- B. `double Calculate(LargeObject* object)`
- C. `double Calculate(LargeObject& object)`

Om `LargeObject` tar upp mycket minne, hur påverkas exekveringstiden för version B jämfört med version A? Snabbare, långsammare eller ingen skillnad? Varför?

Om `LargeObject` tar upp mycket minne, hur påverkas exekveringstiden för version C jämfört med version A? Snabbare, långsammare eller ingen skillnad? Varför? (6p)

2. Jämför funktionerna

```
static int DoSomething(LargeObject object)
int DoSomething(LargeObject object) const
```

Vad skiljer en `const`-funktion och en `static`-funktion (funktionerna är identiska i övrigt)? Kan en funktion vara både `static` och `const`? Förklara varför eller varför inte? (6p)

3. Du ska använda koden för klassen `Vehicle` nedan som basklass för subklassen `Motorcycle`. När du designar `Motorcycle` ser du att du kommer att använda och ändra medlemsvariablerna på många ställen, och du vill kunna göra detta utan att använda `Get`-funktionerna. Nedanstående kod fungerar dock inte detta på det sätt som du skulle vilja. Varför? Hur vill du lösa det och vad får din lösning för konsekvenser? (4p)

```
class Vehicle
{
public:
    Vehicle(int vehicleId, std::string vehicleColor);

    void SetOwner(Person* Owner);
    Person* GetOwner() const;
    int GetVehicleID() const;
    std::string GetVehicleColor() const;

private:
    int m_ID;
    Person* m_Owner;
    std::string m_Color;
};
```

4. Vad är multipelt arv? I kursen tog vi upp ett välkänt problem med multipelt arv. Beskriv detta. (4p)

5. I samband med arv är det viktigt att tänka på minneshantering. Vad är det som är särskilt med minneshantering vid arv och hur löser man det? (2p)