

# KURSBESKRIVNING

---

Objektorienterad Mjukvarudesign  
Object Oriented Software Design  
Kurs på forskarnivå  
Motsvarar 5 högskolepoäng

---

## 1 Syfte

Arkitekturen och designen av ett mjukvarusystem påverkar i hög grad kvaliteten på systemet och kostnaden för utvecklingen. Utgående från grundläggande objektorienterade begrepp och designprinciper analyseras och modelleras ett systems struktur och beteende med hjälp av modelleringsspråket UML (Unified Modelling Language) i en strukturerad arbetsmetodik, Unified Process.

Arbetsmetodiken utgår från en beskrivning av kundkrav och bygger en spårbar kedja via olika UML-modeller hela vägen fram till implementation och testning. Den strukturerade arbetsmetodiken och modelleringsspråket UML ger ett stöd till designarbetet, men för att skapa en hållbar programvarudesign krävs också en förståelse för grundläggande designprinciper och designmönster.

Designmönster är generella lösningsförslag på vanligt förekommande problem, och som mjukvaruutvecklare förväntas man känna till och kunna anpassa dessa generella lösningar till de konkreta utmaningar man försöker lösa. Grundläggande designprinciper beskriver hur man fördelar olika typer av ansvar mellan klasser för att få löst kopplad och lättunderhållen programvarukod.

## 2 Innehåll

Kursen omfattar följande:

- Introduktion till Systemutvecklingsprocessen
- Introduktion till Kravhantering
- Tids- och storleksskattningar
- Planering av arbete
- Grundläggande begrepp inom objektorienterad modellering
- Introduktion till modelleringsspråket UML
- Introduktion till arbetsmetodiken Unified Process
- Grundläggande designprinciper, t.ex. låg koppling, hög sammanhållning, inkapsling, och polymorfism
- Introduktion till, och användning av Designmönster
- Testning, i synnerhet ur mjukvarudesignprocessperspektivet

- Introduktion till Mjukvaruarkitekturer
- Introduktion till Mjukvaruarkitekturstilar

## 3 Mål

### Kunskap och Förståelse

Efter genomförd kurs skall studenten:

- kunna visa kunskap om och förståelse för en systematisk arbetsmetodik
- kunna visa förståelse för grundläggande principer i objektorienterad programvaruutveckling
- kunna visa förståelse för UML som modelleringsspråk
- kunna visa kunskap om grundläggande designprinciper
- kunna visa kunskap om grundläggande designmönster
- kunna visa kunskap om grundläggande mjukvaruarkitekturstilar
- kunna visa förståelse för hur designarbete skapar förutsättningar för testning av mjukvarusystem.

### Färdighet och Förmåga

Efter genomförd kurs skall studenten:

- på en grundläggande nivå kunna ta fram och bedöma krav på en programvara
- på en grundläggande nivå kunna planera och bedöma planer för utvecklingsarbete
- kunna uttrycka strukturen och beteendet hos ett system i termer av objektorienterade koncept
- kunna analysera strukturen och beteendet hos ett system i termer av objektorienterade koncept
- kunna korrekt använda och läsa UML för att uttrycka struktur och beteende hos ett system
- kunna tillämpa grundläggande designprinciper för en objektorienterad design
- kunna tillämpa grundläggande designmönster i en objektorienterad design.
- kunna tillämpa grundläggande arkitekturstilar för ett mjukvarusystem

- kunna resonera om de kvalitetsegenskaper ett system med en viss arkitekturstil har eller bör ha
- kunna resonera om och skapa en grundläggande testplan för ett objektorienterat system.

#### Värderingsförmåga och Förhållningssätt

Efter genomförd kurs skall studenten:

- kunna analysera källkod för eventuella förbättringar
- kunna analysera och kritiskt diskutera en design för eventuella förbättringar.

## 4 Lärande och Undervisning

Undervisningen består av ingående muntliga och skriftliga diskussioner. Kursen ges på engelska.

## 5 Bedömning och Examination

Examinationen består av aktivt deltagande i diskussioner samt av ingående bedömningar av olika, av studenter producerade, projekt-, design-, arkitektur- och testrapporter.

Bedömning i kursen är betyget godkänd/underkänd.

## 6 Kursvärdering

Kursansvarig ansvarar för att forskarstudenten har möjlighet att lämna synpunkter på kursen.

## 7 Kurslitteratur och övriga lärresurser

huvudbok:

- C. Larman, *Applying UML and Patterns*, Prentice Hall, 3rd Edition.

referenslitteratur:

- Gamma, Helm, Johnson, Vlissides, *Design Patterns, Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional.
- L. Bass, R. Kazman, P. Clements, *Software Architecture in Practice*, Addison-Wesley, 2012.
- R. Nystrom, *Game Programming Patterns*, Geneva Benning, 2014.

## 8 Kursansvarig

Mikael Svahnberg  
Blekinge Institute of Technology  
SE-371 79 Karlskrona SWEDEN  
Mikael.Svahnberg@bth.se

