

# Introduktion till Programmering

Dåtid, nutid och framtid

# Kursens syfte

- Att ge en introduktion i programmeringen som hantverk.
  - IDEs, tolkning, kompilering, felsökning, informationsinhämtning, etc etc
- Att lära er att *lära er* att programmera.
  - Möta och hantera hinder.
  - Ni sätter era egna begränsningar.

# Examinering

- Kursen examineras genom
  - deltagande på föreläsningar och övningstillfällen
  - vid kursens slut, uppvisande av ett eller flera fungerande program av lämplig svårighetsgrad

# Bred roll

- Alla behöver inte (bör inte!) bli programmerare!
- ... men alla behöver förstå programmering.

# Om er

- Hur många av er...
  - Har (i grupp) skrivit minst ett fungerande program?
  - Har gått upp på en högre nivå i självskattning?
  - Tycker att programmering är roligt?
  - Har googlat fram svaret på en fråga ni stötte på under kursen?
  - Ser fram emot att möta och överkomma hinder och svårigheter i kommande kurser?
  - Har ett konto på GitHub?

# Livsvisdom I

- Vänta inte på att någon ska ge dig lösningen
  - Du lär dig alltid något på att försöka hitta svaret själv först, även om du inte lyckas.
  - Det finns alltid något att lära sig.
  - Du lär dig mer på att hitta ett svar själv, även om du inte förstår svaret. Att diskutera ett svar du inte förstår med en assistent ger mer än att få svaret serverat.

# Livsvisdom II

- Gör hobbyprojekt – nu!
  - Du lär dig bäst om du gör det i kontexten av ett projekt du brinner för.
  - Vänta inte tills du blir ”färdiglärd” – det blir du aldrig!

# Livsvisdom III

- Vänta inte på "rätt kurs"
  - Om ett ämne intresserar dig, sök information.
  - The internetz are your friend.
  - Läs kursen när det är dags, och få chansen att lära dig ännu mer.
  - Du sätter dina egna begränsningar.
  - Mängden godkända kurser != summan av din kunskap.



# Livsvisdom IV

- Bli aldrig nöjd – fortsätt utmana dig själv
  - Lärandet pågår hela livet – du är aldrig färdig.
  - Vanan att fortsätta lära dig är viktigare än någon specifik kunskap.

# Livsvisdom V

- Lär från varandra, lär ut till varandra
  - ”In learning you will teach –  
in teaching you will learn”
  - Fråga andra om råd och hjälp.
  - Var frikostig med råd och hjälp.

# Vad kommer sen?

- LP1: Introduktion till funktionell programmering
- LP2: Grundläggande datorteknik
- LP3: Objekt-orienterad programmering
- LP4: Programmering av inbyggda system
  
- År2: Datatekniskt projekt, Datastrukturer, Datorsystemteknik, valbart
- År3: (Concurrent programming, Algorithms,) valbart

# Exempel på valbart

- Algoritmer: AI, Machine learning, ...
- Programmering: Advanced functional programming (m fl), Webapplikationer, ...
- Programspråk: Programming language technology, Compiler construction, ...
- Datorsystem: Operativsystem, Real time systems, ...
- Interaktiva simuleringar och spel: Computer graphics, spelprogrammering, ...
- Distribuerade system, Software engineering, Interaktionsdesign, Tillförlitliga system, Datasäkerhet, ...

# Master-program

- CS, Algorithms, Languages and Logic
- Computer Systems and Networks
- Interaktionsdesign
- Embedded Electronic Systems Design
- Software Engineering
- ...

# Kursutvärdering

- Länk kommer att mailas ut – Snälla svara!
- Vi är beroende av att få feedback för att kunna ge bästa möjliga undervisning.
- Positiv och negativ feedback är lika viktig för oss.

# Överlämning

- LP1: Introduktion till funktionell programmering
  - Kursansvarig: Emil Axelsson