

# Programmering 1

## Introduktion

TK22 2023

---

Innehåll

---

Kursens krav

---

Upplägg

---

Läxor & Läxhjälp

---

Setup & installation

---

Varför python?

---

Veckoplanering

---

PyCharm intro

---

# Kursinformation

- 100 poäng
- Kurskod: PRRPRR01
- Betyg: A-F
- Kurslitteratur: Kursolle.se
- Programmeringsspråk: Python
- Program: PyCharm Community edition
- Placeringar: tk22  
      .....

# Kursens krav

- För att få godkänt i kursen måste man klara följande
  - Prov 1: Moment 3-4
  - Prov 2: Moment 5-6
  - Alla inlämningsuppgifter
    - Presentationer
  - Slutuppgiften: GUI-Applikation
  - Presentation muntligt av slutprojekt
- Ha färdigheten att programmera mindre program utan någon tillgång dokumentation eller internet

# Upplägg måndag

- En vanlig lektion är följande
  - 10 min: Genomgång
  - 40 min: Arbete
  - 10 min: Rast
  - 30 min: Arbete

TIMER TILL LEKTION

---

# Upplägg Fredag

- En vanlig lektion är följande
  - 10 min: Genomgång & Frågor
  - 40 min: Arbete & Redovisning

TIMER TILL LEKTION

---

# Läxor & Läxhjälp

- Inga läxor, men ligger du efter är det ditt eget ansvar
- Läxhjälp:
  - Kommer inte kunna hjälpa er men ni kan nog fråga matematikläxhjälp
- Discord kan användas som komplement

# Setup & installation

- Installera Python
- Installera brew
- Installera PyCharm CE (Community Edition)

# Varför python?

- Det är ett relativt lätt språk
- För att exam.net tillåter oss att köra kod på provtillfället
- Vi kommer kunna göra mycket på kort tid

# Centralt innehåll

- Undervisningen i kursen ska behandla följande centrala innehåll:
  - Grundläggande programmering i ett eller flera programspråk varav minst ett av språken är textbaserat.
  - Programmering och dess olika användningsområden ur ett socialt perspektiv inklusive genus, kultur och socioekonomisk bakgrund.
  - Programmeringens möjligheter och begränsningar utifrån datorns funktionssätt.
  - Strukturerat arbetssätt för problemlösning och programmering.
  - Grundläggande kontrollstrukturer, konstruktioner och datatyper.
  - Arbetsmetoder för förebyggande av programmeringsfel, testning, felsökning och rättning av kod.
  - Grundläggande datastrukturer och algoritmer.
  - Gränssnitt för interaktion mellan program och användare.
  - Normer och värden inom programmering, till exempel läsbarhet, dokumentation, testbarhet, rena gränssnitt och nyttan av standard.



# Betygskriterier

## Betyget E

- Eleven formulerar och planerar i samråd med handledare programmeringsuppgifter med pseudokod eller diagramteknik. I planeringen väljer eleven med viss säkerhet kontrollstrukturer, metoder, variabler, datastrukturer och algoritmer som är adekvata för uppgiften. I sin programmering skapar eleven med konsekvent kodningsstil och tydlig namngivning korrekt, strukturerad och enkelt kommenterad källkod med tillfredsställande resultat. Dessutom väljer eleven med viss säkerhet ett uttryckssätt som är anpassat för att på ett tillfredsställande sätt interagera med den avsedda användaren. Elevens färdiga program eller skript är utförda med tillfredsställande resultat i ett eller flera programspråk som är stabila och robusta i program av enkel karaktär.

# Betyget E

- Eleven anpassar med viss säkerhet sin planering av programmeringsuppgiften och utför felsökning av enkla syntaxfel. Innan programmeringsuppgiften avslutas utvärderar eleven med enkla omdömen programmets prestanda och ändamålsenlighet i någon situation och i något sammanhang. Eleven kommunicerar om programmeringsuppgiften och dess utvärdering och använder då med viss säkerhet datavetenskapliga begrepp.
- Eleven redogör översiktligt för programmeringens möjligheter och begränsningar samt hur programmering har påverkat och påverkar vardagen. Eleven redogör översiktligt för principer för att uppnå god kvalitet vid skapandet av datorprogram.

# Betyget D

- Elevens kunskaper bedöms sammantaget vara mellan C och E.

# Betyget C

- Eleven formulerar och planerar efter samråd med handledare programmeringsuppgifter med pseudokod eller diagramteknik. I planeringen väljer eleven med viss säkerhet kontrollstrukturer, metoder, variabler, datastrukturer och algoritmer som är adekvata för uppgiften. I sin programmering skapar eleven med konsekvent kodningsstil och tydlig namngivning korrekt, strukturerad och noggrant kommenterad källkod med tillfredsställande resultat. Dessutom väljer eleven med viss säkerhet ett uttryckssätt som är anpassat för att på ett tillfredsställande sätt interagera med den avsedda användaren. Elevens färdiga program eller skript är utförda med tillfredsställande resultat i ett eller flera programspråk som är stabila och robusta.

# Betyget C

- Eleven anpassar med viss säkerhet sin planering av programmeringsuppgiften och utför på ett systematiskt sätt felsökning av syntaxfel, körtidsfel och programmeringslogiska fel. Innan programmeringsuppgiften avslutas utvärderar eleven med nyanserade omdömen programmets prestanda och ändamålsenlighet i några situationer och sammanhang. Eleven kommunicerar om programmeringsuppgiften och dess utvärdering och använder då med viss säkerhet datavetenskapliga begrepp.
- Eleven redogör utförligt för programmeringens möjligheter och begränsningar samt hur programmering har påverkat och påverkar vardagen. Eleven redogör utförligt för principer för att uppnå god kvalitet vid skapandet av datorprogram.

# Betyget B

- Elevens kunskaper bedöms sammantaget vara mellan A och C.

# Betyget A

- Eleven formulerar och planerar efter samråd med handledare programmeringsuppgifter med pseudokod eller diagramteknik. I planeringen väljer eleven med säkerhet kontrollstrukturer, metoder, variabler, datastrukturer och algoritmer som är adekvata för uppgiften samt motiverar utförligt sina val. I sin programmering skapar eleven med konsekvent kodningsstil och tydlig namngivning korrekt, strukturerad och noggrant och utförligt kommenterad källkod med gott resultat. Dessutom väljer eleven med säkerhet ett uttryckssätt som är anpassat för att på ett gott sätt interagera med den avsedda användaren. Elevens färdiga program eller skript är utförda med gott resultat i ett eller flera programspråk som är stabila och robusta i program av komplex karaktär.

# Betyget A

- Eleven anpassar med säkerhet sin planering av programmeringsuppgiften och utför på ett systematiskt och effektivt sätt felsökning av syntaxfel, körtidsfel och programmeringslogiska fel. Innan programmeringsuppgiften avslutas utvärderar eleven med nyanserade omdömen och med förslag på förbättringar programmets prestanda och ändamålsenlighet i flera situationer och sammanhang. Eleven kommunicerar om programmeringsuppgiften och dess utvärdering och använder då med säkerhet datavetenskapliga begrepp.
- Eleven redogör utförligt och nyanserat för programmeringens möjligheter och begränsningar samt hur programmering har påverkat och påverkar vardagen. Eleven redogör utförligt och nyanserat för principer för att uppnå god kvalitet vid skapandet av datorprogram.



# Veckoplanering

Veckoplaneringen kan ändras, men alla de momenten som räknas upp kommer att behandlas. Det är upp till dig att följa med!

Vecka	Moment	Att göra	Kommentar
v34	Intro	Setup	
v35	<u>Moment02 - Grunderna (Datatyper)</u>	m02u03, m02u04, (Utbyggnad 1, 2)	
v36	<u>Moment02 - Grunderna (Datatyper)</u>	m02u03, m02u04, (Utbyggnad 1, 2)	
v37	<u>Moment03 - Selektioner (if-satser)</u>	Alla, men ej: m03u06	
v38	<u>Moment04 - Iterationer (for-loop)</u>	Alla uppgifter	
v39	<u>Moment05 - Funktioner och datalagring</u>	Alla uppgifter	
v40	<u>Moment06 - Moduler och paket</u>	Alla uppgifter	
v41	<u>P01 Bankapplikation</u>	Slutuppgift	

# Introduction to PyCharm

## What is an IDE?

**Integrated Development Environment (IDE)**

- A software suite that consolidates the basic tools required to write and test software.
- Following chapter will be in English



# Why Do We Need an IDE?

- Streamlines coding with features like:
  - Code completion
  - Syntax highlighting
  - Error detection
- Integrated tools for:
  - Debugging
  - Testing
  - Version control
- Simplifies project management and organization.



# Understanding Python Versions

- Python has multiple versions: 2.x and 3.x.
- Importance of knowing the version:
  - Compatibility
  - Features
  - Libraries
- PyCharm supports multiple Python versions.

Version	Micro version	Release date	End of full support	End of security fixes
3.6	3.6.15[60]	2016-12-23[60]	2018-12-24[b][60]	2021-12-23[60]
3.7	3.7.17[61]	2018-06-27[61]	2020-06-27[b][61]	2023-06-06[61]
3.8	3.8.17[62]	2019-10-14[62]	2021-05-03[b][62]	2024-10[62]
3.9	3.9.17[63]	2020-10-05[63]	2022-05-17[b][63]	2025-10[63][64]
3.10	3.10.12[65]	2021-10-04[65]	2023-04-05[b][65]	2026-10[65]
<b>3.11</b>	3.11.4[66]	2022-10-24[66]	2024-04-01[66]	2027-10[66]
3.12	3.12.0b4[67]	2023-10-02[67]	2025-05[67]	2028-10[67]
3.13	None[68]	2024-10-01[68]	2026-05[68]	2029-10[68]

# How to Check Python Version?

1. Open terminal or command prompt.
2. Type ``python --version`` or ``python3 --version``.
3. The version will be displayed.

 Terminal Python Version



# Where Are My Project Files?

- PyCharm organizes projects in a directory structure.
- Default location varies based on OS.
- Can be customized during project creation.

 PyCharm Project Explorer

# Navigating to Settings

1. Go to ``File` > `Settings`` (Windows/Linux) or ``PyCharm` > `Preferences`` (Mac).
2. Explore categories on the left sidebar.
3. Customize PyCharm to your needs.

 PyCharm Settings