Rybners®
TEKNISKE GYMNASIUM

MINISTERIET FOR BØRN OG UNDERVISNING
KVALITETS- OG TILSYNSSTYRELSEN

Undervisningsbeskrivelse

Stamoplysninger til brug ved prøver til gymnasiale uddannelser

Termin 2023-2024

Institution Rybners Tekniske Gymnasium

Uddannelse HTX

Fag og niveau Programmering B

Lærer(e) Carsten Sørensen

Hold HTX 3D

Oversigt over gennemførte undervisningsforløb

1. år:

Titel 1 - Introduktion til struktureret programmering m. JavaScript

Titel 2 - Programmering/Informatik Tværfagligt Projekt 1

Titel 3 - Grundlæggende struktureret programmering m. Python

Titel 4 - Grundlæggende funktionel programmering m. Python

Titel 5 - Grundlæggende objekt-orienteret programmering m. Python

Titel 6 - Database-programmering m. Python

Titel 7 - Programmering/Informatik Tværfagligt Projekt 2

Titel 8 - Regular Expressions

2. år:

Titel 9 - Designmønstre m. Python

Titel 10 - Spilprogrammering m. Python

Titel 11 - Videregående struktureret programmering m. Java

Titel 12 - Videregående funktionel programmering m. Java

Titel 13 - Grundlæggende objekt-orienteret programmering m. Java

Titel 14 - Hovedopgave 2. år **Titel 15 -**

Studieretningscase

3. år:

Titel 16 - x

Titel 17 - Grafiske brugergrænseflader m. Java

Rybners®
TEKNISKE GYMNASIUM

MINISTERIET FOR BØRN OG UNDERVISNING
KVALITETS- OG TILSYNSSTYRELSEN

Titel 18 - Algoritmer m. Java

Titel 19 - Analyse af selvvalgt sprog

Titel 20 - Eksamensprojekt

1 Introduktion til struktureret programmering m. C#

1.1 Indhold

Kernestof: • Programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

IDE: Visual studio Code

Emner:

Variabler, loops, if/else, switch/case, datatyper, funktioner, array, lister.

https://www.w3schools.com/C#

https://mcronberg.github.io/bogenomcsharp/index.html.

https://www.freecodecamp.org/learn.

1.2 Omfang

6 lektioner

1.3 Særlige fokuspunkter

Kompetencer:

- Redegøre for arkitekturen af programmer på forskellige abstraktionsniveauer, herunder relationen mellem brug og funktion
- Rette, tilpasse og udvide avancerede programmer
- Arbejde inkrementelt og systematisk i programmeringsprocessen
- Demonstrere viden om fagets identitet og metoder

1.4 Væsentligste arbejdsformer

- Tavleundervisning
- Gruppearbejde
- Hands-on projektarbejde
- Opgaveløsning

2 Programmering/Informatik Tværfagligt Projekt 1

2.1 Indhold

Kernestof: • Programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Arkitekturen for programmers interaktion med omgivelserne med henblik på hændelsesstyret interaktion og interaktion mellem systemer
- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding
- Abstrakte programmeringsbeskrivelser og dokumentation

IDE: https://repl.it/languages/html
Emner:
HTML, CSS, webudvikling **Anvendt**
litteratur:
Materialer fra forløb 1
Diverse materialer fra: https://www.w3schools.com/html/, https://www.w3schools.com/css/ Bogen om Informatik v0.7, gennemgået i Informatik.

2.2 Omfang

4 lektioner fordelt mellem fagene

2.3 Særlige fokuspunkter

Kompetencer:

- Bruge programmering til at undersøge et emne eller problemområde, med henblik på – via programmets funktion - at skabe ny indsigt eller til at løse et problem
- Behandle problemstillinger i samspil med andre fag
- Redegøre for arkitekturen af programmer på forskellige abstraktionsniveauer, herunder relationen mellem brug og funktion
- Rette, tilpasse og udvide avancerede programmer
- Arbejde inkrementelt og systematisk i programmeringsprocessen
- Demonstrere viden om fagets identitet og metoder

2.4 Væsentligste arbejdsformer

- Gruppearbejde
- Hands-on projektarbejde

3 Grundlæggende struktureret programmering m. C#

3.1 Indhold

Kernestof: • Programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Arkitekturen for programmers interaktion med omgivelserne med henblik på hændelsesstyret interaktion og interaktion mellem systemer
- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding

IDE: Visual Studio Code
Emner:
Variabler, if/else, switch/case, datatyper, brugerinteraktion **Anvendt litteratur:**
https://www.w3schools.com/C#
https://mcronberg.github.io/bogenomcsharp/index.html

Egenfremstillede kodeeksempler

3.2 Omfang

8 lektioner

3.3 Særlige fokuspunkter

Kompetencer:

- Redegøre for arkitekturen af programmer på forskellige abstraktionsniveauer, herunder relationen mellem brug og funktion
- Redegøre for simple specifikationsmodeller og realisere disse i simple velstrukturerede programmer samt teste disse
- Rette, tilpasse og udvide avancerede programmer
- Arbejde inkrementelt og systematisk i programmeringsprocessen
- Demonstrere viden om fagets identitet og metoder

3.4 Væsentligste arbejdsformer

- Tavleundervisning
- Gruppearbejde
- Opgaveløsning

4 Grundlæggende funktionel programmering m. Python

4.1 Indhold

Kernestof: • programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Arkitekturen for programmers interaktion med omgivelserne med henblik på hændelsesstyret interaktion og interaktion mellem systemer
- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding

IDE: Visual Studio Code
Emner:
Funktioner, lister, tupel, dictionary, loops **Anvendt litteratur:**
https://www.w3schools.com/C#/
https://mcronberg.github.io/bogenomcsharp/index.html

Egenfremstillede kodeeksempler

4.2 Omfang

11 lektioner

4.3 Særlige fokuspunkter

Kompetencer:

- Redegøre for arkitekturen af programmer på forskellige abstraktionsniveauer, herunder relationen mellem brug og funktion
- Redegøre for simple specifikationsmodeller og realisere disse i simple velstrukturerede programmer samt teste disse
- Rette, tilpasse og udvide avancerede programmer
- Arbejde inkrementelt og systematisk i programmeringsprocessen
- Demonstrere viden om fagets identitet og metoder

4.4 Væsentligste arbejdsformer

- Tavleundervisning
- Gruppearbejde
- Hands-on projektarbejde
- Opgaveløsning

5 Objekt-orienteret programmering m. C#

5.1 Indhold

Kernestof: • programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Arkitekturen for programmers interaktion med omgivelserne med henblik på hændelsesstyret interaktion og interaktion mellem systemer
- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding

IDE: Visual Studio Code C#
Klasser, objekter, metoder, nedarvning, constructor, klassediagram, dokumentation **Anvendt litteratur:**
https://www.w3schools.com/cs/cs_oop.php
https://mcronberg.github.io/bogenomcsharp/index.html

5.2 Omfang

10 lektioner

5.3 Særlige fokuspunkter

Kompetencer:

- Redegøre for arkitekturen af programmer på forskellige abstraktionsniveauer, herunder relationen mellem brug og funktion
- Redegøre for simple specifikationsmodeller og realisere disse i simple velstrukturerede programmer samt teste disse
- Anvende avancerede konstruktioner i et programmeringssprog
- Rette, tilpasse og udvide avancerede programmer
- Arbejde inkrementelt og systematisk i programmeringsprocessen
- Demonstrere viden om fagets identitet og metoder
- Abstrakte programmeringsbeskrivelser og dokumentation.

- Tavleundervisning
- Gruppearbejde
- Hands-on projektarbejde

5.4 Væsentligste arbejdsformer

- Opgaveløsning

6 Database-programmering m. C# og SqlLite

6.1 Indhold

Kernestof: • programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Arkitekturen for programmers interaktion med omgivelserne med henblik på hændelsesstyret interaktion og interaktion mellem systemer
- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding
- Generiske programdele og biblioteksmoduler

IDE: Visual Studio Code C# / SQLLite **Emner:**
Databaser (SQLite), databehandling **Anvendt litteratur:**
https://sqlite.org/docs.html
Egen fremstillede kodeeksempler.

6.2 Omfang

6 lektioner

6.3 Særlige fokuspunkter

Kompetencer:

- Redegøre for simple specifikationsmodeller og realisere disse i simple velstrukturerede programmer samt teste disse
- Anvende avancerede konstruktioner i et programmeringssprog
- Rette, tilpasse og udvide avancerede programmer
- Arbejde inkrementelt og systematisk i programmeringsprocessen
- Demonstrere viden om fagets identitet og metoder
- Behandle problemstillinger i samspil med andre fag

- Gruppearbejde

6.4 Væsentligste arbejdsformer

- Hands-on projektarbejde

7 Programmering/Informatik Tværfagligt Projekt 2

7.1 Indhold

Kernestof: • programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Arkitekturen for programmers interaktion med omgivelserne med henblik på hændelsesstyret interaktion og interaktion mellem systemer
- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding
- Generiske programdele og biblioteksmoduler

IDE: Visual Studio Code SQLite m. SQLite Studio
Emner:
Databaser (SQLite), databehandling
Anvendt litteratur:
https://mcronberg.github.io/bogenomcsharp/index.ht
ml
Egenfremstillet introduktion til SQLite Egenfremstillede kodeeksempler

7.2 Omfang

6 lektioner fordelt mellem fagene

7.3 Særlige fokuspunkter

Kompetencer:

- Redegøre for simple specifikationsmodeller og realisere disse i simple velstrukturerede programmer samt teste disse
- Anvende avancerede konstruktioner i et programmeringssprog
- Rette, tilpasse og udvide avancerede programmer
- Arbejde inkrementelt og systematisk i programmeringsprocessen
- Demonstrere viden om fagets identitet og metoder
- Behandle problemstillinger i samspil med andre fag

7.4 Væsentligste arbejdsformer

- Tavleundervisning
- Opgaveløsning

8.4 Væsentligste arbejdsformer

8 Regular Expressions

8.1 Indhold

Kernestof: • Programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Arkitekturen for programmers interaktion med omgivelserne med henblik på hændelsesstyret interaktion og interaktion mellem systemer
- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding
- Abstrakte programmeringsbeskrivelser og dokumentation

Anvendt litteratur:
David Lindholm - Davids bog om Regular Expressions v0.3, Kapitel 2, 3, 6, og 13 Regexr.com

8.2 Omfang

6 lektioner

8.3 Særlige fokuspunkter

Kompetencer:

- Bruge programmering til at undersøge et emne eller problemområde, med henblik på – via programmets funktion - at skabe ny indsigt eller til at løse et problem
- Anvende avancerede konstruktioner i et programmeringssprog
- Redegøre for arkitekturen af programmer på forskellige abstraktionsniveauer, herunder relationen mellem brug og funktion
- Redegøre for simple specifikationsmodeller og realisere disse i simple velstrukturerede programmer samt teste disse
- Arbejde inkrementelt og systematisk i programmeringsprocessen.

8.4 Væsentligste arbejdsformer

- Tavleundervisning
- Gruppearbejde
- Opgaveløsning

9 Designmønstre m. C#

9.1 Indhold

Kernestof: • Programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding
- Abstrakte programmeringsbeskrivelser og dokumentation.

IDE: Visual Studio Code **Emner:**

Singleton, Model View Controller **Anvendt litteratur:**
Egenfremstillede materialer

9.2 Omfang

12 lektioner

9.3 Særlige fokuspunkter

Kompetencer:

- Anvende avancerede konstruktioner i et programmeringssprog
- Demonstrere viden om fagets identitet og metoder

9.4 Væsentligste arbejdsformer

- Tavleundervisning
- Opgaveløsning

11 Videregående struktureret programmering m. Java

11.1 Indhold

Kernestof: • Programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

• Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding

IDE: Visual Studio Code **Emner:**
Variabler, konstanter, datatyper, brugerinput, if/else, switch/case, den ternære operator, kommentarer, while, for, do-while **Anvendt litteratur:**
https://mcronberg.github.io/bogenomcsharp/index.html + Egenfremstillede kodeeksempler

11.2 Omfang

19 lektioner

11.3 Særlige fokuspunkter

Kompetencer:

- Bruge programmering til at undersøge et emne eller problemområde, med henblik på – via programmets funktion - at skabe ny indsigt eller til at løse et problem
- Redegøre for simple specifikationsmodeller og realisere disse i simple velstrukturerede programmer samt teste disse
- Rette, tilpasse og udvide avancerede programmer

11.4 Væsentligste arbejdsformer

- Tavleundervisning
- Gruppearbejde
- Opgaveløsning

12 Videregående funktionel programmering m. Java

12.1 Indhold

Kernestof: • Programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding
- Abstrakte programmeringsbeskrivelser og dokumentation

IDE: Visual studio Code **Emner:**
metoder, polymorfi, rekursion, paradigmer, array, arraylist, linkedlist, kø, stak, Manãna Princippet **Anvendt litteratur:**
https://mcronberg.github.io/bogenomcsharp/index.html + Egenfremstillede kodeeksempler

12.2 Omfang

17 lektioner

12.3 Særlige fokuspunkter

Kompetencer:

- Bruge programmering til at undersøge et emne eller problemområde, med henblik på – via programmets funktion - at skabe ny indsigt eller til at løse et problem
- Anvende avancerede konstruktioner i et programmeringssprog
- Redegøre for simple specifikationsmodeller og realisere disse i simple velstrukturerede programmer samt teste disse
- Rette, tilpasse og udvide avancerede programmer
- Demonstrere viden om fagets identitet og metoder

12.4 Væsentligste arbejdsformer

- Tavleundervisning
- Gruppearbejde
- Opgaveløsning

13 Grundlæggende objekt-orienteret programmering m. C#

13.1 Indhold

Kernestof: • Programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Generiske programdele og biblioteksmoduler
- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding
- Abstrakte programmeringsbeskrivelser og dokumentation

IDE: Visual studio Code **Emner:**
Klasser, objekter, nedarvning, klassediagram, tilstandsdiagram, biblioteker, constructor, accessors, indkapsling, forbindelsestyper og kardinaliteter, command pattern **Anvendt litteratur:**
https://mcronberg.github.io/bogenomcsharp/index.html + Egenfremstillede kodeeksempler

13.2 Omfang

20 lektioner

13.3 Særlige fokuspunkter

Kompetencer:

- Bruge programmering til at undersøge et emne eller problemområde, med henblik på – via programmets funktion - at skabe ny indsigt eller til at løse et problem
- Anvende avancerede konstruktioner i et programmeringssprog
- Redegøre for arkitekturen af programmer på forskellige abstraktionsniveauer, herunder relationen mellem brug og funktion
- Redegøre for simple specifikationsmodeller og realisere disse i simple velstrukturerede programmer samt teste disse
- Rette, tilpasse og udvide avancerede programmer
- Demonstrere viden om fagets identitet og metoder
- Arbejde inkrementelt og systematisk i programmeringsprocessen

13.4 Væsentligste arbejdsformer

- Tavleundervisning
- Gruppearbejde
- Opgaveløsning

14 Hovedopgave 2. år

14.1 Indhold

Kernestof: • Programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Arkitekturen for programmers interaktion med omgivelserne med henblik på hændelsesstyret interaktion og interaktion mellem systemer
- Generiske programdele og biblioteksmoduler
- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding
- Abstrakte programmeringsbeskrivelser og dokumentation.

Emner:
Projektarbejde med IT-system baseret på udleveret oplæg **Anvendt litteratur:** Pensum
Elevernes egen research

14.2 Omfang

14 lektioner

14.3 Særlige fokuspunkter

Kompetencer:

- Bruge programmering til at undersøge et emne eller problemområde, med henblik på – via programmets funktion - at skabe ny indsigt eller til at løse et problem
- Anvende avancerede konstruktioner i et programmeringssprog
- Redegøre for arkitekturen af programmer på forskellige abstraktionsniveauer, herunder relationen mellem brug og funktion
- Redegøre for simple specifikationsmodeller og realisere disse i simple velstrukturerede programmer samt teste disse
- Demonstrere viden om fagets identitet og metoder
- Arbejde inkrementelt og systematisk i programmeringsprocessen.

14.4 Væsentligste arbejdsformer

- Gruppearbejde
- Hands-on projektarbejde

15 Studieretningscase

15.1 Indhold

Kernestof: • Programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Behandle problemstillinger i samspil med andre fag
- Arkitekturen for programmers interaktion med omgivelserne med henblik på hændelsesstyret interaktion og interaktion mellem systemer
- Generiske programdele og biblioteksmoduler
- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding

- Abstrakte programmeringsbeskrivelser og dokumentation.

Emner:
Projektarbejde med IT-system baseret på udleveret oplæg Løsning af tværfaglig opgave med Matematik A.
Anvendt litteratur: Pensum
Elevernes egen research

15.2 Omfang

1 uge

15.3 Særlige fokuspunkter

Kompetencer:

- Bruge programmering til at undersøge et emne eller problemområde, med henblik på – via programmets funktion - at skabe ny indsigt eller til at løse et problem
- Anvende avancerede konstruktioner i et programmeringssprog
- Redegøre for arkitekturen af programmer på forskellige abstraktionsniveauer, herunder relationen mellem brug og funktion
- Redegøre for simple specifikationsmodeller og realisere disse i simple velstrukturerede programmer samt teste disse
- Demonstrere viden om fagets identitet og metoder
- Arbejde inkrementelt og systematisk i programmeringsprocessen.

15.4 Væsentligste arbejdsformer

- Hands-on projektarbejde

16 Videregående objekt-orienteret programmering m. C#

16.1 Indhold

Kernestof: • Programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Generiske programdele og biblioteksmoduler
- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding
- Abstrakte programmeringsbeskrivelser og dokumentation

IDE: Visual Studio.* **Emner:**
Klasser, objekter, nedarvning, klassediagram, abstrakte klasser, interface, enumerator, singleton, versionsstyring **Anvendt litteratur:**
https://mcronberg.github.io/bogenomcsharp/index.html + Egenfremstillede kodeeksempler

16.2 Omfang

15 lektioner

16.3 Særlige fokuspunkter

Kompetencer:

- Bruge programmering til at undersøge et emne eller problemområde, med henblik på – via programmets funktion - at skabe ny indsigt eller til at løse et problem
- Anvende avancerede konstruktioner i et programmeringssprog
- Redegøre for arkitekturen af programmer på forskellige abstraktionsniveauer, herunder relationen mellem brug og funktion
- Redegøre for simple specifikationsmodeller og realisere disse i simple velstrukturerede programmer samt teste disse
- Rette, tilpasse og udvide avancerede programmer
- Demonstrere viden om fagets identitet og metoder
- Arbejde inkrementelt og systematisk i programmeringsprocessen

16.4 Væsentligste arbejdsformer

- Tavleundervisning
- Gruppearbejde
- Opgaveløsning

17 Grafiske brugergrænseflader m. C#

17.1 Indhold

Kernestof: • Programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Arkitekturen for programmers interaktion med omgivelserne med henblik på hændelsesstyret interaktion og interaktion mellem systemer
- Generiske programdele og biblioteksmoduler
- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding
- Abstrakte programmeringsbeskrivelser og dokumentation

IDE: Visual Studio Code **Emner:**
Grafikprogrammering, brugergrænsefladedesign, Windows Forms.
Anvendt litteratur:
https://mcronberg.github.io/bogenomcsharp/index.html + Egenfremstillede kodeeksempler

17.2 Omfang

20 lektioner

17.3 Særlige fokuspunkter

Kompetencer:

- Bruge programmering til at undersøge et emne eller problemområde, med henblik på – via programmets funktion - at skabe ny indsigt eller til at løse et problem
- Anvende avancerede konstruktioner i et programmeringssprog
- Redegøre for arkitekturen af programmer på forskellige abstraktionsniveauer, herunder relationen mellem brug og funktion
- Redegøre for simple specifikationsmodeller og realisere disse i simple velstrukturerede programmer samt teste disse
- Arbejde inkrementelt og systematisk i programmeringsprocessen.

17.4 Væsentligste arbejdsformer

- Tavleundervisning
- Gruppearbejde
- Opgaveløsning

18 Algoritmer m. C#

18.1 Indhold

Kernestof: • Programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding
- Abstrakte programmeringsbeskrivelser og dokumentation

IDE: Visual Studio **Emner:**
Søgning, sortering, tidskompleksitet **Anvendt litteratur:**
https://www.toptal.com/developers/sorting-algorithms
https://mcronberg.github.io/bogenomcsharp/index.html

18.2 Omfang

15 lektioner

18.3 Særlige fokuspunkter

Kompetencer:

- Anvende avancerede konstruktioner i et programmeringssprog
- Redegøre for arkitekturen af programmer på forskellige abstraktionsniveauer, herunder relationen mellem brug og funktion
- Rette, tilpasse og udvide avancerede programmer
- Demonstrere viden om fagets identitet og metoder

18.4 Væsentligste arbejdsformer

- Tavleundervisning
- Opgaveløsning

19 Analyse af selvvalgt sprog

19.1 Indhold

Kernestof:

- Generiske programdele og biblioteksmoduler
- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding • Abstrakte programmeringsbeskrivelser og dokumentation

Emner:
C/C++ ELLER Elm/Ruby on Rails ELLER Fortran/COBOL ELLER Go/Scala ELLER Android **Anvendt litteratur:**
Tutorialsæt til de valgte sprog

19.2 Omfang

20 lektioner

19.3 Særlige fokuspunkter

Kompetencer:

- Bruge programmering til at undersøge et emne eller problemområde, med henblik på – via programmets funktion - at skabe ny indsigt eller til at løse et problem
- Demonstrere viden om fagets identitet og metoder
- Arbejde inkrementelt og systematisk i programmeringsprocessen

19.4 Væsentligste arbejdsformer

- Gruppearbejde
- Hands-on projektarbejde

20 Eksamensprojekt

20.1 Indhold

Kernestof: • Programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre

- Arkitekturen for programmers interaktion med omgivelserne med henblik på hændelsesstyret interaktion og interaktion mellem systemer
- Generiske programdele og biblioteksmoduler
- Arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding
- Abstrakte programmeringsbeskrivelser og dokumentation.

Emner:

Projektarbejde med IT-system baseret på udleveret oplæg **Anvendt litteratur:** Pensum
Elevernes egen research
Eleverne er i forbindelse med projektet blevet introduceret til anvendelse af AI (specifikt ChatGPT) til brug ved fejlfinding, kode review, informationssøgning, og udarbejdelse af kode ud over gruppens evner. Denne anvendelse kan være i forbindelse med projektet eller den tilhørende prøve.
AI genereret kode skal markeres som værende fremstillet af tredjepart, præcist som hvis koden f.eks. kom fra en bog eller hjemmeside.

20.2 Omfang

20 klokketimer

20.3 Særlige fokuspunkter

Kompetencer:

- Bruge programmering til at undersøge et emne eller problemområde, med henblik på – via programmets funktion - at skabe ny indsigt eller til at løse et problem
- Anvende avancerede konstruktioner i et programmeringssprog
- Redegøre for arkitekturen af programmer på forskellige abstraktionsniveauer, herunder relationen mellem brug og funktion
- Redegøre for simple specifikationsmodeller og realisere disse i simple velstrukturerede programmer samt teste disse
- Demonstrere viden om fagets identitet og metoder
- Arbejde inkrementelt og systematisk i programmeringsprocessen.

20.4 Væsentligste arbejdsformer

- Gruppearbejde
- Hands-on projektarbejde