Rybners
TEKNISKE GYMNASIUM

MINISTERIET FOR BØRN OG UNDERVISNING
KVALITETS- OG TILSYNSSTYRELSEN

Undervisningsbeskrivelse

Stamoplysninger til brug ved prøver til gymnasiale uddannelser

Termin	2020-2023
Institution	Rybners Tekniske Gymnasium
Uddannelse	HTX
Fag og niveau	Programmering B
Lærer(e)	David Lindholm
Hold	HX20e

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 - Videregående objekt-orienteret programmering m. Java
Titel 17 - Grafiske brugergrænseflader m. Java
Titel 18 - Algoritmer m. Java
Titel 19 - Analyse af selvvalgt sprog
Titel 20 - Eksamensprojekt

1 Introduktion til struktureret programmering m. JavaScript

1.1 Indhold

Kernestof:

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

IDE: https://repl.it/languages/javascript
Emner:
Variabler, loops, if/else, switch/case, datatyper, funktioner, array
Anvendt litteratur:
https://www.w3schools.com/js/
https://www.freecodecamp.org/learn/ (JavaScript)
Egenfremstillede kodeeksempler

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/, og David Lindholm - 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. Python

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: PyCharm m. Python 3.6.8
Emner:
Variabler, if/else, switch/case, datatyper, brugerinteraktion
Anvendt litteratur:
https://docs.python.org/3/
https://www.w3schools.com/python/
Davids bog om Programmering i Python 3 v0.5, kapitel 1-8
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: PyCharm m. Python 3.6.8
Emner:
Funktioner, lister, tupel, dictionary, loops
Anvendt litteratur:
https://docs.python.org/3/
https://www.w3schools.com/python/
Davids bog om Programmering i Python 3 v0.5, kapitel 9, 10, 13-17
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. Python

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: PyCharm m. Python 3.6.8
Emner:
Klasser, objekter, metoder, nedarvning, constructor, klassediagram, dokumentation
Anvendt litteratur:
https://www.tutorialspoint.com/python/python_classes_objects.htm
https://docs.python.org/3/tutorial/classes.html
Davids bog om Programmering i Python 3 v0.5, kapitel 18-21
Egenfremstillede kodeeksempler

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.

5.4 Væsentligste arbejdsformer

- Tavleundervisning
- Gruppearbejde
- Hands-on projektarbejde
- Opgaveløsning

6 Database-programmering m. Python

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: PyCharm m. Python 3.6.8 & SQLite m. SQLite Studio
Emner:
Databaser (SQLite), databehandling
Anvendt litteratur:
https://sqlite.org/docs.html
https://www.pythoncentral.io/introduction-to-sqlite-in-python/
https://docs.python.org/2/library/sqlite3.html
Egenfremstillet introduktion til SQLite
Egenfremstillede 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

6.4 Væsentligste arbejdsformer

- Gruppearbejde
- 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: PyCharm m. Python 3.6.8 & SQLite m. SQLite Studio
Emner:
Databaser (SQLite), databehandling
Anvendt litteratur:
https://sqlite.org/docs.html
https://www.pythoncentral.io/introduction-to-sqlite-in-python/
https://docs.python.org/2/library/sqlite3.html
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 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. Python

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: PyCharm m. Python 3.6.8 & Tkinter
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

10 Spilprogrammering m. Python

10.1 Indhold

Kernestof:

- 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

IDE: PyCharm m. Python 3.6.8 & Pygame 1.9.4
Emner:
Spiludvikling, spildesign, brugerinteraktion, feedback, grafikprogrammering, basal kunstig intelligens
Anvendt litteratur:
https://nerdparadise.com/programming/pygame/part1
https://pythonprogramming.net/pygame-python-3-part-1-intro/
http://thepythongamebook.com/en:pygame:start
Salen, K & Zimmerman, E. - Rules of Play - Kap 7 - "Defining Games"

10.2 Omfang

26 lektioner

10.3 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 arkitekturen af programmer på forskellige abstraktionsniveauer, herunder relationen mellem brug og funktion
- Anvende avancerede konstruktioner i et programmeringssprog
- Arbejde inkrementelt og systematisk i programmeringsprocessen

10.4 Væsentligste arbejdsformer

- Gruppearbejde
- Hands-on projektarbejde

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: Netbeans m. Java 1.8.*
Emner:
Variabler, konstanter, datatyper, brugerinput, if/else, switch/case, den ternære operator, kommentarer, while, for, do-while
Anvendt litteratur:
David Lindholm - Davids bog om Programmering i Java v0.20, Kapitel 3-7
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: Netbeans m. Java 1.8.*
Emner:
metoder, polymorfi, rekursion, paradigmer, array, arraylist, linkedlist, kø, stak, Manãna Princippet
Anvendt litteratur:
David Lindholm - Davids bog om Programmering i Java v0.20, Kapitel 8-12
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. Java

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: Netbeans m. Java 1.8.*
Emner:
Klasser, objekter, nedarvning, klassediagram, tilstandsdiagram, biblioteker, constructor, accessors, indkapsling, forbindelsestyper og kardinaliteter, command pattern
Anvendt litteratur:
David Lindholm - Davids bog om Programmering i Java v0.20, Kapitel 13-16
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. Java

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: Netbeans m. Java 1.8.*
Emner:
Klasser, objekter, nedarvning, klassediagram, abstrakte klasser, interface, enumerator, singleton, javadoc, versionsstyring
Anvendt litteratur:
David Lindholm - Davids bog om Programmering i Java v0.20, Kapitel 13-16
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. Java

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: Netbeans m. Java 1.8.* & Swing
Emner:
Grafikprogrammering, brugergrænsefladedesign, grafiske elementer, brugerinteraktion, model view controller
Anvendt litteratur:
David Lindholm - Davids bog om Programmering i Java v0.20, Kapitel 18-20, 22
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. Java

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: Netbeans m. Java 1.8.*
Emner:
Søgning, sortering, tidskompleksitet
Anvendt litteratur:
https://www.toptal.com/developers/sorting-algorithms
https://www.geeksforgeeks.org/time-complexities-of-all-sorting-algorithms/
https://www.sortvisualizer.com/
David Lindholm - Davids bog om Programmering i Java v0.20, Kapitel 24-26

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

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