



Undervisningsbeskrivelse

Termin	June 2026
Institution	Himmerlands Erhvervs- og Gymnasieuddannelser
Uddannelse	vaf
Fag og niveau	Programmering B
Lærer	Michael Bohl Jenner (mje)
Hold	3k25

Forløbsoversigt (9)

Forløb 1	Specifikation, algoritmer, pseudokode
Forløb 2	Call stack og rekursion
Forløb 3	Rutediagrammer
Forløb 4	Relationsdatabaser og SQL
Forløb 5	Onlinetræningsplatform (kattis)
Forløb 6	Unity spiludvikling og hændelser
Forløb 7	Objekt Orienteret Programmering
Forløb 8	Eksamensprojekt
Forløb 9	Afslutning og opsamling

Førløb 1: Specifikation, algoritmer, pseudokode

Førløb 1	Specifikation, algoritmer, pseudokode
Indhold	<p>Om udarbejdelse af god specifikation for algoritme. Om arbejde med algoritmer herunder anvendelse af pseudokode. Udarbejde egen applikation som foretager statistiske beregninger (tværfagligt).</p> <p>Noter: Opdater IDE (Visual Studio og Rider) til nyeste IKKE-beta. Prompt AI til at give dig pseudokoden med samme format som InsertionSort() i noten, men for algoritmen for specifikation om at summere tal, se noten her. Lav note om følgende, hvor du beskriver med egne ord. Giv eksempler på programmeringstekniske problemstillinger jævnfør at vi arbejder med algoritmer. Hvordan stiller man en specifikation op (til løsning af problem)? Hvad er en algoritme? Hvad er pseudokode for noget? Hvilket pseudokodeformat bruger vi? Forklar en komplet opbygning af en metode (aka funktion). Hvordan tester man en implementation af en pseudokode (dvs. kode). Mvh Michael Løs en delopgave mere i førløbs-noten om algoritmer. Se og forstå visualiseringen af FindMax algoritme her. Se og forstå visualiseringen af InsertionSort algoritmen her. Lav en pseudokode for en af de algoritmer der skal indgå i projektet "Statistik med .NET". Viser til lærer på forespørgsel.</p>
Omfang	12 lektioner / 12 timer
Særlige fokuspunkter	<p>Fagmål: behandle problemstillinger i samspil med andre fag 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</p> <p>Kernestof: programmeringssprog og elementer i programmets opbygning, herunder variable, 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</p>
Væsentligste arbejdsformer	Primært individuelt arbejde med mulighed for sparring.

Forløb 2: Call stack og rekursion

Forløb 2	Call stack og rekursion
Indhold	<p>Call stack og rekursion. Live demo med visning af call stack i Visual Studio. Øvelser med små programmer hvor call stack monitoreres (også i Rider / macOS). Opgaver med udarbejdelse af call stack på papir. Opgaver med løsning af små rekursionsalgoritmer på papir.</p> <p>Noter: Tag en dialog med en AI maskine om hvad rekursion er i programmering? om der er fordele og ulemper ved at anvende rekursion? Skriv kortfattet note med dine egne ord, med de 5 væsentligste pointer om rekursion, fordele og ulemper. Løs en opgave ekstra i udleverede opgaveark? Hvad er en "call stack"? Forklares med egne ord. Hvad er en stak-ramme? Forklares med egne ord. Her er lidt noter fra sidst, note1 og note2. Hvad mener jeg med at "en funktion kalder sig selv", i virkeligheden burde formuleres "en funktion kalder en instans af sig selv"? Hvis ovenstående er vanskelige, så tag meget gerne en dialog med chatbot om emnerne. Løs resten af opgaverne i de to opgavesæt om hhv call stack og om rekursion. Skriv note i din logbog om call stack og om rekursion.</p>
Omfang	5 lektioner / 5 timer
Særlige fokuspunkter	<p>Fagmål: 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</p> <p>Kernestof: programmeringssprog og elementer i programmers opbygning, herunder variable, 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</p>
Væsentligste arbejdsformer	Individuelt arbejde med mulighed for sparring.

Forløb 3: Rutediagrammer

Forløb 3	Rutediagrammer
Indhold	Miniforløb om udarbejdelse af rutediagrammer. Noter: Løs en opgave i sættet om rutediagram, som aftalt i timen i dag.
Omfang	1 lektion / 1 timer
Særlige fokuspunkter	Kernestof: programmeringssprog og elementer i programmers opbygning, herunder variable, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre abstrakte programmeringsbeskrivelser og dokumentation
Væsentligste arbejdsformer	Individuelt med mulighed for sparring.

Forløb 4: Relationsdatabaser og SQL

Forløb 4	Relationsdatabaser og SQL
Indhold	<p>Overordnet teori om relationsdatabaser og SQL. Ambitionen er ikke at eleverne kan forstå simple database schema og systemer med en relation.</p> <p>Kort information om normalformer og formålet med disse.</p> <p>Introduktion til anvendelse af SQLite og Dapper (ORM).</p> <p>Introduktion til ERD diagrammer.</p> <p>Udarbejdelse af egen applikation med benspænd: der skal indgå database med mindst en relation.</p> <p>Noter:</p> <p>Vi starter på forløb om SQL databaser.</p> <p>Løs opgaver frem til og med afsnit 3.</p> <p>Opdatering: Dette dokument indeholder samme information som de næste to, men er pænere formateret. (Instruks: Læs om database notation, diagrammer og design-trin her.). Læs om database-notation, database-diagrammer (ERD) og design-trin. Her er fil med korrekte symboler til diagrammer. Kun de første 10 sider skimmes. Fokus: Hvordan kan du skrive vores system med Drink og Kategori (2-tabel-system) med kort notation. (F.eks. når AI skal promptes). Hvad er forskellen på 1-til-mange relation og mange-til-mange relation i forhold til hvordan det indbygges i et relations-database system bestående af flere tabeller?</p> <p>Læs om skema-notation til effektiv beskrivelse af et databasesystem (gør det nemmere at prompte AI). Læs om rammerne for projekt med databaser.</p> <p>Forbered demo af og forklaring af statistik i .net projektet. Tidsramme: 5 minutter. I timen: fremlæggelse af statistik-projekter. Dernæst videre med projektbeskrivelse til næste projekt (med databaser).</p>
Omfang	12 lektioner / 12 timer
Særlige fokuspunkter	<p>Fagmål:</p> <p>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</p> <p>behandle problemstillinger i samspil med andre fag</p> <p>anvende avancerede konstruktioner i et programmeringssprog</p> <p>redegøre for arkitekturen af programmer på forskellige abstraktionsniveauer, herunder relationen mellem brug og funktion</p> <p>redegøre for simple specifikationsmodeller og realisere disse i simple velstrukturerede programmer samt teste disse</p> <p>rette, tilpasse og udvide avancerede programmer</p> <p>demonstrere viden om fagets identitet og metoder</p> <p>arbejde inkrementelt og systematisk i programmeringsprocessen</p> <p>Kernestof:</p> <p>programmeringssprog og elementer i programmers opbygning, herunder variable, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre</p> <p>generiske programdele og biblioteksmoduler</p> <p>arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding</p> <p>abstrakte programmeringsbeskrivelser og dokumentation</p>

Væsentligste arbejdsformer	Lærerstyret undervisning. Gruppearbejde.
---------------------------------------	---

Førløb 5: Onlinetræningsplatform (kattis)

Førløb 5	Onlinetræningsplatform (kattis)
Indhold	<p>Introduktion til træningsplatformen kattis. Herunder streams: stdin, stdout og stderr. Dernæst vælger elever selv sværhedsgrader og træner at komme fra specifikation til løsning.</p> <p>Noter: Læs introduktion til kattis problem-løsning, dels her, og dels ved at søge hjælp på deres hjemmeside.</p>
Omfang	7 lektioner / 7 timer
Særlige fokuspunkter	<p>Fagmål: 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 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</p> <p>Kernestof: programmeringssprog og elementer i programmers opbygning, herunder variable, 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</p>
Væsentligste arbejdsformer	Lærerstyret undervisning med live coding. Individuelt elevarbejde med differentierede træningsopgaver.

Forløb 6: Unity spiludvikling og hændelser

Forløb 6	Unity spiludvikling og hændelser
Indhold	<p>Kort introduktion til Unity. Introduktion til opbygning af typisk spilengine og rollerne af Update() og Start() i Unity. Arbejde med Unity-tutorial Flappy Birds. Hændelser og events. Udarbejdelse af eget 2D spil i Unity.</p> <p>Noter: Installer Unity på din bærbare, så du kan afprøve "at kode i Unity" i et forløb om Unity. Opret personlig konto hos Unity (vi har prøvet skolekonto det er noget rod, som andre skoler jeg har talt med om dette heller ikke bruger). Hvis du ikke allerede har gjort det så: Installer Unity på din bærbare, så du kan afprøve "at kode i Unity" i et forløb om Unity. Opret personlig konto hos Unity (vi har prøvet skolekonto det er noget rod, som andre skoler jeg har talt med om dette heller ikke bruger). Unity-spil-projekt. Læs om event i programmering. Fokus: hvad er udgiver (publisher) og abonnent (subscriber)? Fokus: hvordan abonnerer et objekt på en event i C#? Fokus: hvordan udgiver et objekt et event i C# Hvad er en eventhandler? Hvordan bruger følgende system events? Opdater din logbog med erfaringer fra Unity-arbejdet.</p>
Omfang	19 lektioner / 19 timer
Særlige fokuspunkter	<p>Fagmål: 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 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</p> <p>Kernestof: programmeringssprog og elementer i programmers opbygning, herunder variable, 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</p>
Væsentligste arbejdsformer	Gruppearbejde.

Forløb 7: Objekt Orienteret Programmering

Forløb 7	Objekt Orienteret Programmering
Indhold	<p>Objekt-orienteret programmering. Planlæg undervisningsforløb om OOP til 2g&#39;ere. Udarbejdelse af klassesdiagram. Derefter OOP opgavesæt. Udarbejd egen Learning App, ud fra (guided) vejledning.</p> <p>Noter: Meld dig på skoletubekanal for holdet: https://www.skoletube.dk/group/3k25ProgrammeringHEGHTX/. Brug join koden: 9C4283D3. Opstart på forløb om OOP. I OOP har forskellige objekter behov for at kende hinanden. Skriv en tekst på 3-5 linjer om hvordan objekter kan kommunikere. Du må sparre med AI, men skal skrive det med egne ord. Hav teksten klar til aflevering. Inspiration: Hvad er OOP. Metoder og instanser. Referencer. Constructor. Access modifiers. Læs desuden om at fremstille klassesdiagrammer, som er en oplagt dokumentationsform når man har OOP. Bemærk: Målet er at i kan fremstille korrekt relationsdiagram blot med linjer og evt pil for læse-retning med "use", "know", "is-a", "has-a", "owns-a" tekst på, så ser vi bort fra om korrek piltype er på eller ej. Det er unødvendige detaljer - fordi vi fokuserer på OOP ikke OOD. Lav klassesdiagram over dit unity-spil. Dokumentation om klassesdiagrammer findes her. Vælg en tilfældig blandt de 12 eksamens-øve-opgaveeksempler her (som du ikke har arbejdet med torsdag). Gennemgå koden så du kan forklare hvad den gør. Du må sparre med AI, men prøv først uden AI, men med de hjælpemidler der findes her (side med link til lovligt materiale til forberedelsestid). I forhold til OOP: Hvad er en constructor? Giv eksempler. Se videorække om OOP, her. Skim datatyper her: https://gymdok.jenner.dk/progny/datatyper.html. Aflevering af opgave 1, tilfældigt trukket fra eksamenslignende sæt. Undersøg svarende på følgende: Hvorfor er der forskel på hvor meget hukommelse en ascii-tegn tager og en char i C#? Giv kode-eksempel hvor du lagrer tegn i byte-styrrelse. Hvad er metode-overloading? Giv kode-eksempler. Hvad er polymorfi? Giv kode-eksempler. Hav besvarelserne klar i word dokument som du kan aflevere efter gennemsyn.</p>
Omfang	23 lektioner / 23 timer

<p>Særlige fokuspunkter</p>	<p>Fagmål: bruge programmering til at undersøge et emne eller problemområde, med henblik på ζ 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</p> <p>Kernestof: programmeringssprog og elementer i programmers opbygning, herunder variable, 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</p>
<p>Væsentligste arbejdsformer</p>	<p>Individuelt arbejde med opgavesæt og learning app. Vejledning er use, modify, create-opbygget.</p>

Forløb 8: Eksamensprojekt

Forløb 8	Eksamensprojekt
Indhold	<p>Eksamensprojekt.</p> <p>Noter: Se video om at fremstille rutediagram (for use case) og om at fremstille klassediagram. Casen er til .NET MAUI app. Video hedder rutediagram-klassediagram... Videoen blev lidt lang, så skimning af indhold er ok. Interessant video om AI skiftet.</p>
Omfang	20 lektioner / 20 timer
Særlige fokuspunkter	<p>Fagmål: 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 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</p> <p>Kernestof: programmeringssprog og elementer i programmers opbygning, herunder variable, 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</p>
Væsentligste arbejdsformer	<p>Gruppearbejde til udvikling af produkt. Individuelt arbejde ved udarbejdelse af synopsen.</p>

Forløb 9: Afslutning og opsamling

Forløb 9	Afslutning og opsamling
Indhold	Afslutning og opsamling. Noter: Forbered evt mundtlig eksamen. Vi laver produkt-og-synopsis-præsentationer.
Omfang	5 lektioner / 5 timer
Væsentligste arbejdsformer	Lærerstyret dialog.