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

Forløb 1: Specifikation, algoritmer, pseudokode

Forløb 1	Specifikation, algoritmer, pseudokode
Indhold	Om udarbejdelse af god specifikation for algoritme. Om arbejde med al- goritmer herunder anvendelse af pseudokode. Udarbejde egen applikati- on som foretager statistiske beregninger (tværfagligt). Noter: Opdater IDE (Visual Studio og Rider) til nyeste IKKE-beta. Prompt AI til at give dig pseudokoden med samme format som InsertionS- ort() 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? Hvilk- et pseudokodeformat bruger vi? Forklar en komplet opbygning af en meto- de (aka funktion). Hvordan tester man en implementation af en pseudoko- de (dvs. kode). Mvh Michael Løs en delopgave mere i forløbs-noten om algoritmer. Se og forstå visu- aliseringen 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". Vises til lærer på forespørgsel.
Omfang	12 lektioner / 12 timer
Særlige fokuspunkter	Fagmål: behandle problemstillinger i samspil med andre fag anvende avancerede konstruktioner i et programmeringssprog redegøre for arkitekturen af programmer på forskellige abstraktionsniv- eauer, 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 Kernestof: programmeringssprog og elementer i programmers opbygning, herunder var- iabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktions- mekanismer, rekursion, polymorfi og algoritmemønstre generiske programdele og biblioteksmoduler arbejdsgange og systematik i programmeringsprocessen, herunder test og fejlfinding abstrakte programmeringsbeskrivelser og dokumentation
Væsentligste arbejdsformer	Primært individuelt arbejde med mulighed for sparring.

Forløb 2: Call stack og rekursion

Forløb 2	Call stack og rekursion
Indhold	Call stack og rekursion. Live demo med visning af call stack i Visu- al Studio. Øvelser med små programmer hvor call stack monitoreres (og- så i Rider / macOS). Opgaver med udarbejdelse af call stack på pap- ir. Opgaver med løsning af små rekursionsalgoritmer på papir. 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 kortfatt- et note med dine egne ord, med de 5 væsentligste pointer om rekursi- on, 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 fu- nktion kalder en instans af sig selv"? Hvis ovenstående er vanskelig- e, så tag meget gerne en dialog med chatbot om emnerne. Løs resten af opgaverne i de to opgavesæt om hhv call stack og om reku- rsion. Skriv note i din logbog om call stack og om rekursion.
Omfang	5 lektioner / 5 timer
Særlige fokuspunkter	Fagmål: anvende avancerede konstruktioner i et programmeringssprog redegøre for arkitekturen af programmer på forskellige abstraktionsniv- eauer, herunder relationen mellem brug og funktion rette, tilpasse og udvide avancerede programmer demonstrere viden om fagets identitet og metoder Kernestof: programmeringssprog og elementer i programmers opbygning, herunder var- iabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktions- mekanismer, 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
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 variabler, 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	Overordnet teori om relationsdatabaser og SQL. Ambitionen er ikke at eleverne kan forstå simple database schema og systemer med en relation. Kort information om normalformer og formålet med disse. Introduktion til anvendelse af SQLite og Dapper (ORM). Introduktion til ERD diagrammer. Udarbejdelse af egen applikation med benspænd: der skal indgå database med mindst en relation. Noter: Vi starter på forløb om SQL databaser. Løs opgaver frem til og med afsnit 3. 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? Læs om skema-notation til effektiv beskrivelse af et databasesystem (gør det nemmere at prompte AI). Læs om rammerne for projekt med database. 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).
Omfang	12 lektioner / 12 timer
Særlige fokuspunkter	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 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

Væsentligste arbejdsformer	Lærerstyret undervisning. Gruppearbejde.

Forløb 5: Onlinetræningsplatform (kattis)

Forløb 5	Onlinetræningsplatform (kattis)
Indhold	Introduktion til træningsplatformen kattis. Herunder streams: stdin, stdout og stderr. Dernæst vælger elever selv sværhedsgrader og træner at komme fra specikation til løsning. Noter: Læs introduktion til kattis problem-løsning, dels her, og dels ved at søge hjælp på deres hjemmeside.
Omfang	7 lektioner / 7 timer
Særlige fokuspunkter	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 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
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	Kort introduktion til Unity. Introduktion til opbygning af typisk spi- lengine og rollerne af Update() og Start() i Unity. Arbejde med Unity- -tutorial Flappy Birds. Hændelser og events. Udarbejdelse af eget 2D spil i Unity. 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 sko- le-konto 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 bærbar- e, så du kan afprøve "at kode i Unity" i et forløb om Unity. Opret per- sonlig konto hos Unity (vi har prøvet skole-konto 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 ab- onnent (subscriber)? Fokus: hvordan abonnerer et objekt på en event i C#? Fokus: hvordan udgiver et objekt et event i C# Hvad er en eventhan- dler? Hvordan bruger følgende system events? Opdater din logbog med erfaringer fra Unity-arbejdet.
Omfang	19 lektioner / 19 timer
Særlige fokuspunkter	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 abstraktionsniv- eauer, 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 Kernestof: programmeringssprog og elementer i programmers opbygning, herunder var- iabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktions- mekanismer, 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
Væsentligste arbejdsformer	Gruppearbejde.

Forløb 7: Objekt Orienteret Programmering

Forløb 7	Objekt Orienteret Programmering
Indhold	Objekt-orienteret programmering. Planlæg undervisningsforløb om OOP til 2g'ere. Udarbejdelse af klassediagram. Derefter OOP opgaves- æt. Udarbejd egen Learning App, ud fra (guided) vejledning. Noter: Meld dig på skoletubekanalen 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 klassediagrammer, 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 klassediagram over dit unity-spil. Dokumentation om klassediagrammer 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.
Omfang	23 lektioner / 23 timer

Særlige fokuspunkter	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 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 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
Væsentligste arbejdsformer	Individuelt arbejde med opgavesæt og learning app. Vejledning er use, modify, create-opbygget.

Forløb 8: Eksamensprojekt

Forløb 8	Eksamensprojekt
Indhold	Eksamensprojekt. 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 bliv lidt lang, så skimning af indhold er ok. Interessant video om AI skiftet.
Omfang	20 lektioner / 20 timer
Særlige fokuspunkter	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 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
Væsentligste arbejdsformer	Gruppearbejde til udvikling af produkt. Individuelt arbejde ved udarbejdelse af synopsen.

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.

Undervisningsbeskrivelse

Termin	June 2025
Institution	Himmerlands Erhvervs- og Gymnasieuddannelser
Uddannelse	vaf
Fag og niveau	Programmering B
Lærer	Michael Bohl Jenner (mje)
Hold	2k24

Forløbsoversigt (5)

Forløb 1	Introduktion til C# med CLI og array
Forløb 2	GUI med .NET MAUI
Forløb 3	Robotarm og Arduino-web-api
Forløb 4	Frit projekt
Forløb 5	Årsprøveprojekt

Forløb 1: Introduktion til C# med CLI og array

Forløb 1	Introduktion til C# med CLI og array
Indhold	Introduktion til programmering generelt. Introduktion til udviklingsv- ærktøj: Visual Studio Introduktion til sprog generelt og C# konkret. Anvender C# kompendium som har indbyggede opgaver (baseret på nemprog- rammerings C# kursus) Delforløb som viser, ændrer og skaber CLI appl- ikation (use-modify-create) Delforløb som viser, ændrer og skaber konsol applikation der anvender array (use-modify-create). Noter: Installer Visual Studio Community ved at følge vejledningen i starten af vedhæftede pdf. Bemærk: Visual Studio 2022 community kræver cirka 10GB ledig harddisk-plads, meget afhængigt af hvilke "workloads" vi væ- lger at arbejde med. Så inden du går igang med at downloade anbefales det at sikre at du har mindst 12GB ledig plads. Læs de første 11 sider af noten om programmeringssprog mm. Sørg for at du har følgende: 1. Ved hvordan du via string-array args kan indlæse parametre fra kommandoprompten, og lav dit første cliT- est.cs program (første eksempel i noten). 2. Kan opsætte IDE (Visual Studio eller Rider) i et projekt til at overføre kommandolinje paramet- re når man kører programmet via play-knappen (grøn pil). 3. Kan opsætte IDE i et projekt til at kopiere output til din "cli"-folder, f.eks. C:\Users\mje\cli på Windows eller \Users\michael\cli på macOS. Skriv note om hvordan man gør (afleveres hvis der spørges om det). Vejledn- ing: http://gymdok.jenner.dk/prog/forl%C3%B8b-CommandLineInterface-CLI- -mje.pdf Du har nu mindst et CLI program. Kopier det til OneNote klassenotesbog- en "2k24 prog", sektionen Projekter, til siden CLI (som allerede find- es). Note: Det skal kopieres så det er læsbart, anvend Visual Studio Code til at åbne projektets Program.cs fil, og "toggle light" så du an- vender "Light theme" inden du kopierer så der ikke kommer sort baggr- und på teksten (cltr+shift+p, tast toggle light, hvorefter option dukk- er op så du kan skifte). Kør den igen for at skifte tilbage til mørk (mildere ved øjnene). Beskriv derefter, med egne ord, hvordan koden vi- rker, instruktion for instruktion. OneNote klassenotesbogen findes via Teams 2k24 prog, eller via onenote.com. Giv lyd i beskedsystemet hvis det driller. Løs to ekstra delopgaver i C# kompendiet. Se video 21 om array og video 22 om gennemløb af array. Løs opgaverne 1-3 i noten om array.
Omfang	15 lektioner / 15 timer

Særlige fokuspunkter	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 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 Kernestof: programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre generiske programdele og biblioteksmoduler
Væsentligste arbejdsformer	Lærerstyret undervisning. Individuelt arbejde ved opgaveforløb. Gruppearbejde med udleveret CLI forløb (som indeholder inkrementel udvikling, og use-modify-create).

Forløb 2: GUI med .NET MAUI

Forløb 2	GUI med .NET MAUI
Indhold	Udvikling af GUI programmer med .NET MAUI. Introduktion til XAML til layout, hændelsesstyret programmering og til Model-View-ViewModel arkitekturen. Introducerer .NET MAUI via tre små applikationer: Dri- kkepengeberegner. CodeQuotes HangMan Dernæst vælger eleverne selv applikation som udvikles i .NET MAUI. Noter: Læs noten om XAML og skim Microsofts hjemmeside om XAML. Løs alt i forløbet drikkepengeberegninger pånær sidste sektion (hvor du selv skal komme med løsninger. Er du igennem drikkepenge-beregner opgaverne? Hvis ikke løses de sidste.
Omfang	13 lektioner / 13 timer
Særlige fokuspunkter	Fagmål: 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 arbejde inkrementelt og systematisk i programmeringsprocessen 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
Væsentligste arbejdsformer	Lærerstyret undervisning, herunder show the process. Gruppearbejde.

Forløb 3: Robotarm og Arduino-web-api

Forløb 3	Robotarm og Arduino-web-api
Indhold	På grund af begrænset antal robotarme deles eleverne i to grupper, halvdelen arbejder med robot-arm programmering og den anden halvdel med arduino web-api. Efterfølgende bytter eleverne projekt. Emner for robotarm: (a) hvilke to koordinatsystem-sæt Dobot arbejder med, (b) hvad Pose er. (c) hvad jogging og point-to-point styring er. (d) hvad bevægelserne JUMP, MOVL, MOVJ gør. (e) hvad HOMING er. Emner for web-api IP adresser DNS HTTP API HTTP GET, PUT, POST Noter: Læs om Robotter generelt, og om Dobot Magician specielt, i noten med fokus på styring fra Arduino Mega 2560. Når du har skimmet teksten ved du: (a) hvilke to koordinatsystem-sæt Dobot arbejder med, (b) hvad Pose er. (c) hvad jogging og point-to-point styring er. (d) hvad bevægelserne JUMP, MOVL, MOVJ gør. (e) hvad HOMING er. Forbered 7 minutters præsentation af dit sidste projekt (før robot/-esp32). Herunder demo og neddyk i kode. Indtal din præsentation i video, upload til youtube og aflever ved at aflevere link til youtube video. Afleveres i Teams klassenotesbogen i undersiden under Projekter, kaldet "Link til video om .NET MAUI GUI app" IoT er et begreb om "Internet of Things", altså at mange af vores "dimser" skal på Internet. Besvar spørgsmål i din Logbog (findes typisk via Teams, Klassenotesbog, sektionen Logbog) på den side der hedder "60 - Internet of Things". Se evt gymdok siden eller tavlenoter.
Omfang	14 lektioner / 14 timer

Særlige fokuspunkter	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 rette, tilpasse og udvide avancerede programmer demonstrere viden om fagets identitet og metoder 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
Væsentligste arbejdsformer	Lærerstyret undervisning. Individuelt arbejde ved opgaveforløb. Gruppearbejde med udleverede forløb (robot + esp32, som begge indeholder inkrementel udvikling, og use-modify-create).

Forløb 4: Frit projekt

Forløb 4	Frit projekt
Indhold	Frit projekt. Eleverne udvælger selv applikation som de ønsker at udvikle. Det kan være konsol, gui, robot, arduino. Noter: Læs noten om at forklare kode.
Omfang	9 lektioner / 9 timer
Særlige fokuspunkter	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 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
Væsentligste arbejdsformer	Gruppearbejde.

Forløb 5: Årsprøveprojekt

Forløb 5	Årsprøveprojekt
Indhold	Produkt udvikles i grupper på 1-3 personer. Synopsis udarbejdes i gru- pper (denne gang, ikke til rigtig eksamen). Afsluttes med mundtlig ek- samination. Noter: Hej, Jeg har nu oprettet et lækkert lille HTML dokument som giver kort- fattet overblik over hvad i skal kunne i programmering til årsprøve. Tjek det ud og giv gerne feedback. Det findes her. Mvh Michael Planlæg en 10 minutters præsentation af hvad du har nu, og følg vejled- ningen: http://gymdok.jenner.dk/proggen/praesentation.html
Omfang	20 lektioner / 20 timer
Særlige fokuspunkter	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 abstraktionsniv- eauer, 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 Kernestof: programmeringssprog og elementer i programmers opbygning, herunder var- iabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktions- mekanismer, 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
Væsentligste arbejdsformer	Gruppearbejde ved produktudvikling. Individuel skrevet synopsis.