

You need to enable JavaScript to run this app.

Undervisningsbeskrivelse

| | |
|----------------------|-------------------------|
| Termin | maj-juni, 2026 |
| Institution | 280951 |
| Uddannelse | VALGFAG |
| Fag og niveau | 07497 B Programmering B |
| Lærer(e) | Gorm Drachmann |
| Hold | htx225pro |

Oversigt over gennemførte undervisningsforløb i faget

| | |
|-----------------|---|
| Forløb 1 | APP lab |
| Forløb 2 | Grundlæggende programmering i C# |
| Forløb 3 | Database |
| Forløb 4 | WEB:Hjemmesider grundlæggende - ASP.NET |
| Forløb 5 | Gamedevelopment - Unity |

Beskrivelse af det enkelte undervisningsforløb

| | |
|-----------------------------------|--|
| Forløb 1 | APP lab 10 lektioner |
| Forløbets indhold og fokus | Forstå struktur af kode. Se hvordan det fungerer. Drag drop |
| Faglige mål | <ul style="list-style-type: none"> • 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 • Demonstrere viden om fagets identitet og metoder • Arbejde inkrementelt og systematisk i programmeringsprocessen. |
| Kernestof | <ul style="list-style-type: none"> • Programmeringssprog og elementer i programmers opbygning, herunder variabler, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmemønstre |
| Anvendt materiale. | |
| Arbejdsformer | |

| | |
|--|--|
| <p>Forløb 2</p> | <p>Grundlæggende programmering i C# 55 lektioner</p> |
| <p>Forløbets indhold og fokus</p> | <p>Følger Nemprogrammering.dk uv videoer i sproget C#. Der laves opgaver og følges op på tavlen.</p> |
| <p>Faglige mål</p> | <ul style="list-style-type: none"> • 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 • Demonstrere viden om fagets identitet og metoder • Arbejde inkrementelt og systematisk i programmeringsprocessen. |
| <p>Kernestof</p> | <ul style="list-style-type: none"> • 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 |
| <p>Anvendt materiale.</p> | |
| <p>Arbejdsformer</p> | |

| | |
|--|--|
| <p>Forløb 3</p> | <p>Database 10 lektioner</p> |
| <p>Forløbets indhold og fokus</p> | <p>Bruger Microsoft SQL Server Management Studio i forbindelse med brugen af database. Der vil være fokus på følgende:</p> <ul style="list-style-type: none"> - MVC struktur - Oprette tabeller - Hente og gemme data - Hente og gemme data vha VS - Connection strings + Entiteter - Filtrering af data ud fra SQL / Linq - ER diagrammer - Relational database – relationelle forhold mellem tabeller. - JQuery og generiske lister. <p>Perioden afsluttes med en opgave hvor eleverne skal lave et program i WPF der går ind og bruger en lokal database eller database på server.</p> |
| <p>Faglige mål</p> | <ul style="list-style-type: none"> • 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>Kernestof</p> | <ul style="list-style-type: none"> • Programmeringssprog og elementer i programmers opbygning, herunder variable, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmestrukturer • 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>Anvendt materiale.</p> | |
| <p>Arbejdsformer</p> | |

| | |
|--|--|
| <p>Forløb 4</p> | <p>WEB:Hjemmesider grundlæggende - ASP.NET 28 lektioner</p> |
| <p>Forløbets indhold og fokus</p> | <p>Fokusområder: Tager udgangspunkt i et ASP.NET projekt der er bygget op omkring en Module View Controle struktur.</p> <ul style="list-style-type: none"> - Grundlæggende HTML - Tabeller, formularer, og medier - CSS - Dynamisk webpage bla via Bootstrap. - Javascript – Grundlæggende om variabler, funktioner, Loops, betingelser osv. - Webservices, API, perfomance og sikkerhed |
| <p>Faglige mål</p> | <ul style="list-style-type: none"> • 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>Kernestof</p> | <ul style="list-style-type: none"> • 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. |
| <p>Anvendt materiale.</p> | |
| <p>Arbejdsformer</p> | |

| | |
|--|--|
| <p>Forløb 5</p> | <p>Gamedevelopment - Unity 28 lektioner</p> |
| <p>Forløbets indhold og fokus</p> | <p>Differentieret undervisning hvor elever der har valgt gamedevelopment kører et selvstændigt forløb i miljøet Unity. Eleven sætter sig ind i programmet Unity og laver opgaver til områderne. Forløbet er meget selvstudie.</p> |
| <p>Faglige mål</p> | <ul style="list-style-type: none"> • 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 specificationsmodeller 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>Kernestof</p> | <ul style="list-style-type: none"> • Programmeringssprog og elementer i programmers opbygning, herunder variable, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmestrukturer • 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>Anvendt materiale.</p> | |
| <p>Arbejdsformer</p> | |

You need to enable JavaScript to run this app.

Undervisningsbeskrivelse

| | |
|----------------------|-------------------------|
| Termin | maj-juni, 2026 |
| Institution | 280951 |
| Uddannelse | VALGFAG |
| Fag og niveau | 07497 B Programmering B |
| Lærer(e) | Gorm Drachmann |
| Hold | htx325pro |

Oversigt over gennemførte undervisningsforløb i faget

| | |
|-----------------|--|
| Forløb 1 | WEB: Hjemmesider grundlæggende - ASP.NET |
| Forløb 2 | Gamedevelopment - Unity |
| Forløb 3 | WEB : API tjenester - webscrabe |
| Forløb 4 | Før eksamensopgave 1- Unity og WEB |
| Forløb 5 | FørEksamensopgave 2 |
| Forløb 6 | Eksamensprojekt: WEB og Unity |

Beskrivelse af det enkelte undervisningsforløb

| | |
|--|--|
| <p>Forløb 1</p> | <p>WEB: Hjemmesider grundlæggende - ASP.NET 26 lektioner</p> |
| <p>Forløbets indhold og fokus</p> | <p>Fokusområder: Tager udgangspunkt i et ASP.NET projekt der er bygget op omkring en Module View Controle struktur.</p> <ul style="list-style-type: none"> - Grundlæggende HTML - Tabeller, formularer, og medier - CSS - Dynamisk webpage bla via Bootstrap. - Javascript – Grundlæggende om variabler, funktioner, Loops, betingelser osv. - Webservices, API, performance og sikkerhed |
| <p>Faglige mål</p> | <ul style="list-style-type: none"> • 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>Kernestof</p> | <ul style="list-style-type: none"> • 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. |
| <p>Anvendt materiale.</p> | |
| <p>Arbejdsformer</p> | |

| | |
|--|--|
| <p>Forløb 2</p> | <p>Gamedevelopment - Unity 26 lektioner</p> |
| <p>Forløbets indhold og fokus</p> | <p>Differentieret undervisning hvor elever der har valgt gamedevelopment kører et selvstændigt forløb i miljøet Unity. Eleven sætter sig ind i programmet Unity og laver opgaver til områderne. Forløbet er meget selvstudie.</p> |
| <p>Faglige mål</p> | <ul style="list-style-type: none"> • 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 specificationsmodeller 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>Kernestof</p> | <ul style="list-style-type: none"> • Programmeringssprog og elementer i programmers opbygning, herunder variable, typer, udtryk, kontrolstrukturer, parametrisering/abstraktionsmekanismer, rekursion, polymorfi og algoritmestrukturer • 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>Anvendt materiale.</p> | |
| <p>Arbejdsformer</p> | |

| | |
|-----------------------------------|--|
| Forløb 3 | WEB : API tjenester - webscrabe |
| Forløbets indhold og fokus | <p>Tager udgangspunkt i et ASP.NET projekt der er bygget op omkring en Module View Controle struktur.</p> <ul style="list-style-type: none"> - API tjenester - webscrabe - Grundlæggende HTML - Tabeller, formularer, og medier - CSS - Dynamisk webpage bla via Bootstrap. - Javascript – Grundlæggende om variabler, funktioner, Loops, betingelser osv. - Webservices, API, perfomance og sikkerhed |
| Faglige mål | <ul style="list-style-type: none"> • 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 | <ul style="list-style-type: none"> • 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. |
| Anvendt materiale. | |
| Arbejdsformer | |

| | |
|--|--|
| <p>Forløb 4</p> | <p>Før eksamensopgave 1- Unity og WEB 26 lektioner</p> |
| <p>Forløbets indhold og fokus</p> | <p>Et selvstændigt projekt i Unity og WEB. Eleverne følger op på det de har lært. Derudover vil forløbet indeholde:</p> <ul style="list-style-type: none"> - Dokumentation af et IT projekt. - Introduktion til Rekursive funktioner - Polymorfi - Algoritmer |
| <p>Faglige mål</p> | <ul style="list-style-type: none"> • 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>Kernestof</p> | <ul style="list-style-type: none"> • 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>Anvendt materiale.</p> | |
| <p>Arbejdsformer</p> | |

| | |
|-----------------------------------|--|
| Forløb 5 | FørEksamensopgave 2 32 lektioner |
| Forløbets indhold og fokus | Selvstændigt elevprojekt i Unity og WEB. Eleverne følger op på kode. |
| Faglige mål | <ul style="list-style-type: none"> • 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 specificationsmodeller 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 | <ul style="list-style-type: none"> • 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. |
| Anvendt materiale. | |
| Arbejdsformer | |

| | |
|-----------------------------------|--|
| Forløb 6 | Eksamensprojekt: WEB og Unity 26 lektioner |
| Forløbets indhold og fokus | Elevernes selvstændige eksamensprojekt |
| Faglige mål | <ul style="list-style-type: none"> • 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 specificationsmodeller 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 | <ul style="list-style-type: none"> • 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. |
| Anvendt materiale. | |
| Arbejdsformer | |