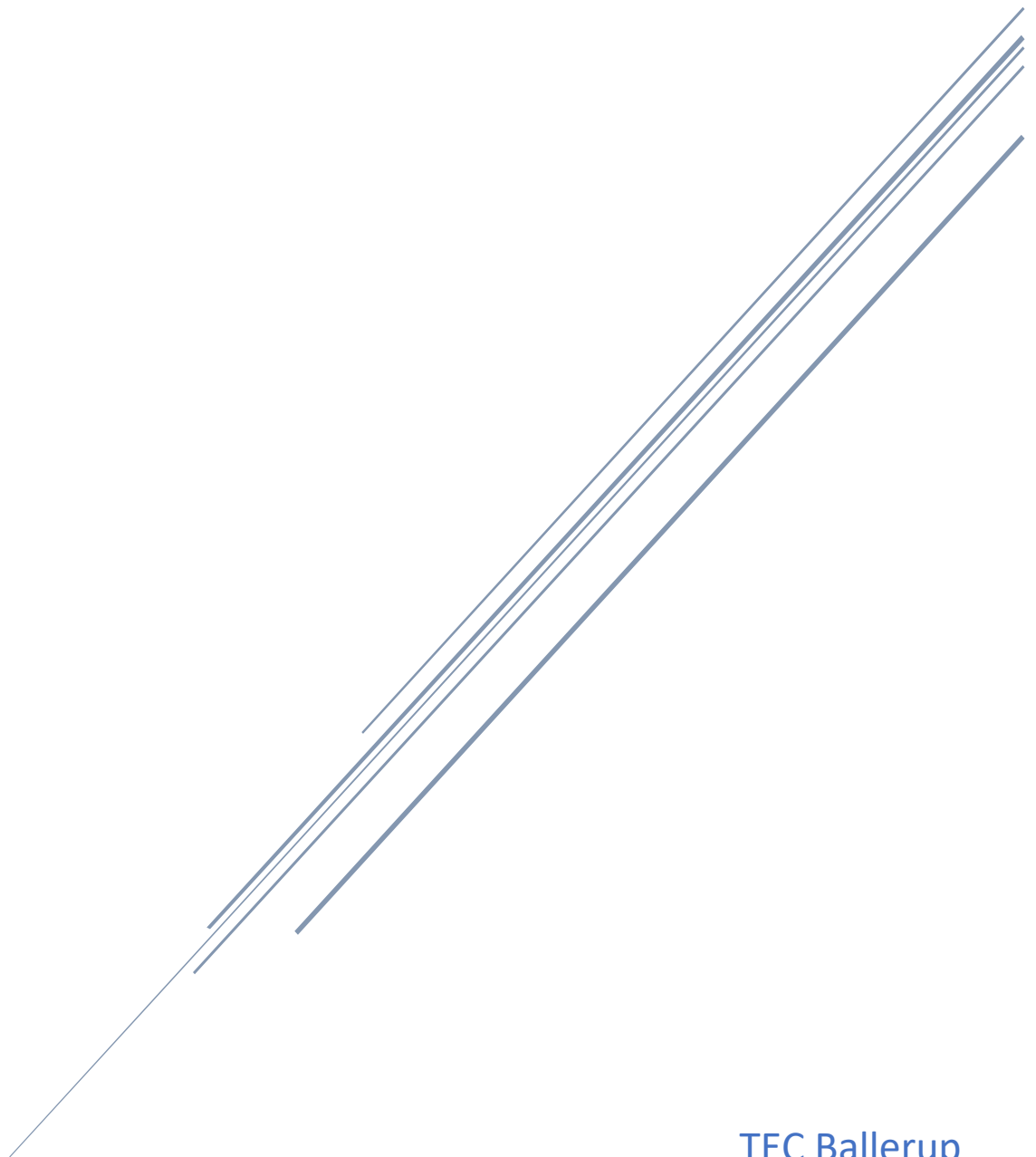


OPGAVESÆT

Grundlæggende Programmering H1



TEC Ballerup
Datateknikker med speciale i programmering

Indholdsfortegnelse

A Begynderopgaver	3
1. Hej Verden Program	3
2. Simpel Variabel Opgave	3
3. Grundlæggende Matematiske Operationer	3
4. If-else Struktur med Alder	3
5. Grader Fahrenheit til Celsius	3
6. Lommeregner	3
7. Brugerdefinerede Metoder	3
8. Loop Eksempel – For Løkke	3
9. Simple Arrays	4
10. Multi-dimensionelle Arrays	4
11. Konstanter	4
12. E-mail Validering	4
13. Fejlretning	4
14. Opretning og Brug af Kommentarer	4
15. Tal Rækkefølge	4
16. Grundlæggende BMI-beregner	4
17. String Formattering	4
18. Talgenkendelse med Løkke	5
19. Grundning af Versionsstyring	5
20. Basis Login System	5
B Rutinerede Opgaver	5
1. Avanceret Lommeregner	5
2. List Sortering	5
3. Palindrome Checker	5
4. Valutakonvertering	5
5. Gæt-nummer-spil med Tilbagekobling	5
6. Pagination af Kontakter	6
7. Brugerdefinerede Fejlmeddelelser	6
8. Matematiske Ligninger	6
9. Bankkonto-Simulator	6
10. Opretning af Diagrammer for et Program	6
11. Logik-baseret Login	6

12. Analysering af Tekstfiler	6
13. Versionering af Programændringer	6
14. Geometriske Beregninger	6
15. Udvidet Månedsbudget.....	6
16. Filter for Tal Rækkefølge.....	7
17. Sorter Kontakter Alfabetisk.....	7
18. E-mail Validering med Regex.....	7
19. Fibonacci Sekvens	7
20. Opgavekommentarer og Dokumentation	7
C Rutinerede Opgaver	8
1. Avanceret Telefonbog med Søge-, Sorterings- og Filtreringsfunktioner	8
2. Personalehåndtering og Lønberegner	8
3. Avanceret Budgetplanlægger og Udgiftssporing	9
4. Biblioteksstyring med Ud- og Indlånsfunktionalitet.....	10
5. Krydsord Generator.....	10

A Begynderopgaver

Disse opgaver er designet til at introducere lærlingen til grundlæggende programmering, datatyper, metoder og simpel fejlretning.

1. Hej Verden Program

- Udskriv "Hej Verden" i konsollen.
- **Fokus:** Introduktion til konsolprogrammering.

2. Simpel Variabel Opgave

- Deklarér en string-variabel med dit navn og udskriv den.
- **Fokus:** Variabler og datatyper.

3. Grundlæggende Matematiske Operationer

- Brugeren indtaster to tal, og programmet udfører addition, subtraktion, multiplikation og division.
- **Fokus:** Enkle operationer, metoder og variabler.

4. If-else Struktur med Alder

- Brugeren indtaster sin alder, og programmet afgør, om de er myndige.
- **Fokus:** Kontrolstrukturer.

5. Grader Fahrenheit til Celsius

- Brugeren indtaster en temperatur i Fahrenheit, og programmet konverterer den til Celsius.
- **Fokus:** Matematiske operationer og datatyper.

6. Lommeregner

- Brugeren vælger en operation og to tal; programmet udfører operationen og viser resultatet.
- **Fokus:** Kontrolstrukturer og metoder.

7. Brugerdefinerede Metoder

- Opret en metode, der udskriver en velkomstbesked, og kald metoden.
- **Fokus:** Metoder uden returværdi.

8. Loop Eksempel – For Løkke

- Programmet udskriver tallene fra 1 til 10.
- **Fokus:** For-løkker og iteration.

9. Simple Arrays

- Lav et program, der udskriver elementerne i en enkel array (f.eks. en liste over frugter).
- **Fokus:** Arrays.

10. Multi-dimensionelle Arrays

- Lav en 2x2 matrix, der repræsenterer en simpel tabel over tal, og udskriv indholdet.
- **Fokus:** Multidimensionelle arrays.

11. Konstanter

- Brug en konstant til at repræsentere moms (f.eks. 25%) og beregn moms for en given pris.
- **Fokus:** Variabler og konstanter.

12. E-mail Validering

- Programmet spørger brugeren om en e-mail og validerer, om den indeholder "@" og ".".
- **Fokus:** String-manipulation og betingelser.

13. Fejlretning

- Giv lærlingen en fejlbehæftet kode og lad dem finde og rette fejlene.
- **Fokus:** Debugging og fejlhåndtering.

14. Opretning og Brug af Kommentarer

- Giv et simpelt program og bed lærlingen kommentere koden.
- **Fokus:** Brug af kodekommentarer.

15. Tal Rækkefølge

- Brugeren indtaster tre tal, og programmet viser dem i stigende rækkefølge.
- **Fokus:** Arrays og kontrolstrukturer.

16. Grundlæggende BMI-beregner

- Brugeren indtaster højde og vægt, og programmet beregner BMI.
- **Fokus:** Matematiske operationer og variabler.

17. String Formattering

- Brugeren indtaster navn og alder, og programmet formaterer og udskriver disse.
- **Fokus:** String manipulation.

18. Talgenkendelse med Løkke

- Brugeren indtaster tal mellem 1 og 10 indtil de gætter det "hemmelige" tal.
- **Fokus:** Løkker og betingelser.

19. Grunding af Versionsstyring

- Forklar versionsstyring og introducer praktisk brug i opgaverne.
- **Fokus:** Versionsstyring (teori og basal praksis).

20. Basis Login System

- Brugeren indtaster brugernavn og adgangskode, og programmet bekræfter adgang, hvis input stemmer.
 - **Fokus:** Variabler, kontrolstrukturer og fejlhåndtering.
-

B Rutinerede Opgaver

Disse opgaver tilføjer kompleksitet og introducerer mere avancerede metoder, arrays, og brug af debugging.

1. Avanceret Lommeregner

- Tilføj modulus og eksponentiering til lommeregneren fra begynderniveau.
- **Fokus:** Kontrolstrukturer, metoder.

2. List Sortering

- Programmet tager en liste med tilfældige tal og sorterer dem.
- **Fokus:** Arrays og kontrolstrukturer.

3. Palindrome Checker

- Programmet tjekker, om en given sætning er et palindrom.
- **Fokus:** String-manipulation og arrays.

4. Valutakonvertering

- Konverter beløb mellem to valutaer ved brug af metoder med parametre.
- **Fokus:** Metoder med returverdier og variabler.

5. Gæt-nummer-spil med Tilbagekobling

- Giv brugeren hints som "højere" eller "lavere" i spillet.
- **Fokus:** Betingelser og debugging.

6. Pagination af Kontakter

- Del en liste over kontakter op i sider (10 pr. side).
- **Fokus:** Arrays og kontrolstrukturer.

7. Brugerdefinerede Fejlmeddelelser

- Programmet skal håndtere og vise tilpassede fejlmeddelelser.
- **Fokus:** Fejlhåndtering.

8. Matematiske Ligninger

- Løs kvadratiske ligninger, fx $ax^2 + bx + c = 0$.
- **Fokus:** Matematik og metoder.

9. Bankkonto-Simulator

- Brugeren kan hæve og indsætte penge, og saldoen opdateres.
- **Fokus:** Metoder og kontrolstrukturer.

10. Opretning af Diagrammer for et Program

- Giv lærlingen en kode, og bed dem tegne et flowdiagram.
- **Fokus:** Diagrammering.

11. Logik-baseret Login

- Brugeren indtaster forkert tre gange, før adgangen blokeres.
- **Fokus:** Fejlhåndtering og kontrolstrukturer.

12. Analysering af Tekstfiler

- Programmet læser en tekstfil og viser antal ord.
- **Fokus:** Filhåndtering og arrays.

13. Versionering af Programændringer

- Lærlingen gemmer ændringer af opgaven med versioneringskommentarer.
- **Fokus:** Versionsstyring.

14. Geometriske Beregninger

- Beregn areal og omkreds af forskellige figurer.
- **Fokus:** Metoder med parametre.

15. Udvidet Månedsbudget

- Brug en array for hver måned og beregn et gennemsnit.

- **Fokus:** Arrays og kontrolstrukturer.

16. Filter for Tal Rækkefølge

- Brugeren indtaster tal, og programmet viser dem i faldende rækkefølge.
- **Fokus:** Arrays og kontrolstrukturer.

17. Sorter Kontakter Alfabetisk

- Sortér en liste af kontakter alfabetisk.
- **Fokus:** Arrays og debugging.

18. E-mail Validering med Regex

- Lav en mere præcis e-mail validering.
- **Fokus:** Regex og string-manipulation.

19. Fibonacci Sekvens

- Vis Fibonacci-sekvensen op til et givet antal.
- **Fokus:** Rekursion og løkker.

20. Opgavekommentarer og Dokumentation

- Kommentér og dokumenter hvert trin i en større opgave.
- **Fokus:** Kommentarer og fejlhåndtering.

C Rutinerede Opgaver

1. Avanceret Telefonbog med Søge-, Sorterings- og Filtreringsfunktioner

Beskrivelse:

Lærlingen skal oprette en telefonbogs-applikation, der giver brugeren mulighed for at tilføje, søge efter, redigere, slette og sortere kontakter. Telefonbogen skal lagres i en tekstfil, og applikationen skal indlæse og gemme dataene ved opstart og lukning.

Funktionalitet:

- **Brugersinput:** Brugeren skal kunne indtaste og redigere kontaktdetaljer (navn, telefonnummer, e-mail).
- **Datahåndtering:** Brug forskellige datatyper (string til navn og e-mail, int eller long til telefonnummer).
- **Metoder:** Implementér metoder til at tilføje, slette, redigere og søge efter kontakter.
- **Returværdier:** Metoder skal returnere specifikke beskeder (f.eks. "Kontakt ikke fundet"), og en liste af kontakter til søgning og filtrering.
- **Kontrolstrukturer:** Brug if-else til at validere data og switch til at navigere i menuen.
- **Arrays og List:** Brug en List<> til at gemme kontakter og implementer søgning og sortering ved hjælp af Linq-metoder som .Where() og .OrderBy().
- **Filhåndtering:** Brug StreamWriter og StreamReader til at gemme og indlæse kontakter fra en tekstfil.
- **Pagination:** Implementer pagination for visning af 10 kontakter ad gangen.

Ekstra:

- **Validering:** Valider indtastede oplysninger som e-mail og telefonnummerformat.
- **Debugging:** Brug debugging-værktøjer til at overvåge og justere dataflows.
- **Kommentarer:** Kommentér koden for at forklare strukturen og metoderne.

2. Personalehåndtering og Lønberegner

Beskrivelse:

Lav en applikation, der kan oprette, redigere og slette medarbejderposter og beregne månedsløn og skattebidrag ud fra forskellige faktorer som ansættelsestype, løn og fradrag. Brugeren kan tilføje oplysninger som navn, arbejdstimer, timeløn og skatteprocent.

Funktionalitet:

- **Brugersinput:** Indlæs oplysninger om medarbejderens navn, timer, timeløn og skatteprocent.

- **Datahåndtering:** Brug forskellige datatyper (int til timer, double til løn og skatteprocent, string til navn).
- **Metoder:** Opret metoder til at beregne løn efter skat, løn før skat, ferietillæg, osv.
- **Returværdier:** Returnér beregnede værdier (nettoløn, bruttoløn) og opdel det i oversigtspunkter.
- **Arrays og List:** Brug en List<> til at gemme medarbejderoplysninger og implementér funktioner til at søge efter medarbejdere.
- **Kontrolstrukturer:** Brug if-else og switch til at differentiere mellem ansættelsestyper og skatteprocenter.
- **Metoder med parametre:** Lav metoder, der accepterer parametre som timeløn, arbejdstimer og skatteprocent.

Ekstra:

- **Versionsstyring:** Implementer versionsstyring for at kunne rulle tilbage til tidligere beregningsversioner.
- **Diagrammering:** Skitser et diagram, der viser hvordan lønberegningen udføres.
- **Debugging:** Brug debugging-værktøjer til at identificere fejl i beregningerne.

3. Avanceret Budgetplanlægger og Udgiftssporing

Beskrivelse:

Opret en budgetplanlægningsapplikation, der hjælper brugeren med at indtaste deres månedlige indtægter og udgifter, kategorisere dem og give en oversigt over månedens samlede økonomiske status. Programmet skal kunne vise historik og lave en prognose for de næste 6 måneder.

Funktionalitet:

- **Brugerinput:** Brugeren skal indtaste indkomst og udgifter opdelt på kategorier som "Mad", "Transport", "Bolig" osv.
- **Datahåndtering:** Brug double til at repræsentere beløb og string til kategorier.
- **Metoder:** Implementer metoder til at beregne total indkomst og udgift, samt metoder til prognose for kommende måneder.
- **Returtyper:** Returnér beløbsspecifikke værdier som budgetstatus, overskud/underskud og kategorispecifikke forbrug.
- **Kontrolstrukturer:** Brug kontrolstrukturer til at beregne månedens økonomiske status og give anbefalinger.
- **Arrays og List:** Brug arrays til at gemme udgifter per kategori, og en List<> til at håndtere historik og prognoser.
- **Pagination og Filhåndtering:** Inddel udgiftshistorik og fremtidig prognose pr. måned og gem i en tekstfil.

Ekstra:

- **Grafik:** Vis en enkel grafisk repræsentation (tekstbaseret) af udgiftshistorik.
 - **Kommentarer:** Kommentér koden for at forklare de økonomiske beregninger.
 - **Diagrammering:** Lav et diagram, der viser kategorierne og deres indflydelse på budgettet.
-

4. Biblioteksstyring med Ud- og Indlånsfunktionalitet

Beskrivelse:

Lav en applikation til at administrere udlån og indlån af bøger i et bibliotek. Brugeren skal kunne søge efter en bog, se tilgængelighed, låne og returnere bøger. Oplysninger om bøgerne gemmes i en tekstfil, og programmet viser, hvem der har udlånt bøgerne.

Funktionalitet:

- **Brugersinput:** Brugeren indtaster bogtitler, ISBN og låneroplysninger.
- **Datahåndtering:** Brug string til titler og låneroplysninger og int eller long til ISBN.
- **Metoder:** Implementer metoder til at låne og returnere bøger samt til at søge efter bøger.
- **Returtyper:** Returnér specifikke meddelelser om tilgængelighed og lånestatus.
- **Kontrolstrukturer:** Brug if-else strukturer til at tjekke, om bøger er tilgængelige, og om en bruger kan låne flere bøger.
- **Arrays og List:** Brug en List<> til at gemme bøger og udlånsoplysninger.

Ekstra:

- **Fejlhåndtering:** Implementer fejlhåndtering for at sikre, at brugeren ikke låner mere end tilladt.
 - **Versionsstyring:** Versionsstyr bogsamlingen og udlånshistorikken.
 - **Kommentarer:** Kommentér koden for at forklare de enkelte metoder og kontrolstrukturer.
-

5. Krydsord Generator

Beskrivelse:

Lav en krydsordsgenerator, hvor brugeren kan vælge niveau (nem, medium, svær) og få en tilfældig genereret krydsord. Brugeren skal kunne løse krydsordet og få feedback på korrekthed og status for opgaveløsningen.

Funktionalitet:

- **Brugerininput:** Brugeren vælger sværhedsgrad og svarer på krydsordspørgsmål.
- **Datahåndtering:** Brug arrays eller lister til at gemme ord og hints.
- **Metoder:** Implementer metoder til at generere krydsord, validere brugerininput og vise feedback.
- **Returtyper:** Returnér svarbeskeder som korrekt/ukorrekt og krydsordløsningens status.
- **Kontrolstrukturer:** Brug kontrolstrukturer til at tjekke sværhedsgrad og validere svar.
- **Arrays og multidimensionelle arrays:** Brug multidimensionelle arrays til at gemme krydsordets struktur og ordplacering.

Ekstra:

- **Debugging:** Brug debugging-værktøjer til at sikre korrekt opsætning af krydsordet.
- **Diagrammering:** Diagrammer krydsordets opbygning og løsningens flow.
- **Versionsstyring:** Versioner krydsordmodulerne og brug historik til at analysere forbedringer.