

DANMARKS TEKNISKE UNIVERSITET

Skriftlig prøve, 14. december 2018, 4 timer

Side 1 af 18

Kursus navn: 02101– Indledende Programmering

Kursus nr.: 02101

Tilladte hjælpemidler: Ikke-digitale skriftlige hjælpemidler er tilladt.

Lommeregner er **ikke** tilladt. Bøger, noter, regnestok, lineal, målebånd, ludobrik, etc. **er** tilladt.

Vejledende vægtning af opgaverne:

Opgave 1: 24 %

Opgave 2: 8 %

Opgave 3: 22 %

Opgave 4: 24 %

Opgave 5: 12 %

Opgave 6: 10 %

Vægtningen er vejledende. Bedømmelsen vil være baseret på en helhedsvurdering.

Alle opgaver besvares ved at udfylde de dertil indrettede felter på de følgende sider. Som opgavebesvarelse afleveres blot disse sider i udfyldt stand. Hvis der opstår pladsmangel kan man eventuelt benytte ekstra papir som vedlægges opgavebesvarelsen.

Bemærk at der står opgaver på begge sider af papiret.

Husk bordnummer, navn, studienummer, fødselsdato og arknummer på side 1.

Husk studienummer og arknummer på samtlige ark.

Bord nr.	Kursus nr.: 02101	Dato: 14. december 2018	Ark nr.
	Kursusnavn: Indledende Programmering		
	Studienr.: _____	Fødselsdato: _____	
	Navn: _____		

Opgave 1 (Multiple choice spørgsmål) 24%

Besvar spørgsmålene givet nedenfor ved for hvert af spørgsmålene at markere netop de rigtige svar med kryds. Der kan være nul, et, eller flere rigtige svar. For at besvare spørgsmålet korrekt, skal alle de rigtige svar gives et kryds, og ingen af de forkerte svar må få et kryds.

Der er 10 spørgsmål af denne type.

I alle spørgsmål benyttes flertalsbøjningen for at reflektere, at der er tale om ubekendt antal. Dvs. f.eks. "hvilke" skal tolkes som "hvilken eller hvilke hvis nogen".

1. Betragt følgende stykke kode:

```
Scanner s = new Scanner(System.in);
if (metode(s) > 6) {
    System.out.println("Bingo");
}
```

Hvilke af følgende typer er lovlige returtype for metode?

- String
- Scanner
- int
- Point
- ArrayList<Point>

2. Betragt følgende stykke kode:

```
int a = 10;
int b = 100;
a = a + b/2;
b = b - a + b/5;
```

Efter eksekvering af ovenstående kode, hvilke af følgende (matematiske) udsagn er da sande om a og b?

- a = 55
- b < 0
- a = 60
- b = a
- a = 10
- b = 0
- b = 100

Bord
nr.

Kursus nr.: 02101

Dato: 14. december 2018

Ark nr.

Kursusnavn: Indledende Programmering

Studienr.: _____ Fødselsdato: _____

Navn: _____

3. Betragt følgende stykke kode:

```
for (int i = 0; i < X; i++) {  
    for (int j = i; j < Y; j++) {  
        System.out.print(i);  
    }  
}
```

Hvilke af følgende output kan for passende valg af X og Y blive skrevet til System.out af koden ovenfor?

- 12345
- 01234
- 000112
- 001123
- 0000111
- 0000111223
- 000000000000000000

4. Betragt følgende stykke kode:

```
if (a.length > 0 && a[0] > 0) {  
    System.out.println("Bingo");  
}
```

For hvilke værdier af a vil koden hverken give compiler- eller runtimefejl?

- int[] a = {20,30,40};
- int[] a = new int[0];
- int[] a = new int[10];
- String[] a = "Hello world".split();

Bord
nr.

Kursus nr.: 02101

Dato: 14. december 2018

Ark nr.

Kursusnavn: Indledende Programmering

Studienr.: _____ Fødselsdato: _____

Navn: _____

5. Betragt følgende stykke kode:

```
Random terning = new Random();  
System.out.println(terning.nextInt(6)+terning.nextInt(6)+2);
```

Programmet bruger tilfældighed, og gør derfor ikke altid det samme. Hvilke af følgende tal *kan* programmet skrive til `System.out`?

- | | |
|--------------------------|--------------------------|
| <input type="radio"/> 0 | <input type="radio"/> 6 |
| <input type="radio"/> 5 | <input type="radio"/> 7 |
| <input type="radio"/> 12 | <input type="radio"/> 14 |

6. Betragt følgende stykke kode:

```
int [] a = {20, 3, 17, 5, 0};  
int [] b = {20, 3, 17, 5, 0};  
int [] c = b;  
Arrays.sort(c);
```

Hvilke af følgende udsagn vil efter eksekvering af ovenstående kode være sande?

- a er et sorteret array
- b er et sorteret array
- c er et sorteret array
- b er null
- c er null

Bord
nr.

Kursus nr.: 02101

Dato: 14. december 2018

Ark nr.

Kursusnavn: Indledende Programmering

Studienr.: _____ Fødselsdato: _____

Navn: _____

7. Betragt følgende stykke kode:

```
while (a.length < 10 || a[0] < 10) {
    for (int i = 0; i < a.length; i++) {
        a[i]++;
    }
}
```

For hvilke værdier af `a` vil programmet gå i uendelig løkke?

- `int[] a = {20, 30, 40};`
- `int[] a = new int[20];`
- `int[] a = {-1,2,-1,2,5,16,18,15,8,7,13,20,-3,30,117,115,112};`
- `int[] a = new int[0];`

8. Betragt følgende stykke kode:

```
public class D<T> {
    T a;
    T b;

    public D(T x, T y) {
        a = x;
        b = y;
    }

    public D() {
        a = b;
    }
}
```

Hvilke af følgende kommandoer instantierer et objekt af typen `D`?

- `D d = new D("new, 1");`
- `D<String> = new D<String>("a","b");`
- `D<String> d = new D<String>(null,null);`
- `D<int> d = new D<int>(4,2);`
- `D<Double> d = new D<Double>(4.2,"16");`
- `D<Object> d = new D();`

Bord
nr.

Kursus nr.: 02101

Dato: 14. december 2018

Ark nr.

Kursusnavn: Indledende Programmering

Studienr.: _____ Fødselsdato: _____

Navn: _____

9. Betragt følgende stykker kode:

```
public class OpgA {
    private String name;
    private int number;
    public OpgA(String s) {
        name = s;
        number = -1;
    }
    public OpgA(int n) {
        number = n;
    }
    public OpgA(String s, int n) {
        name = s;
        number = n;
    }
    public String toString() {
        return name + " " + number;
    }
}

public class OpgB extends OpgA {
    public OpgB(String s, int n) {
        super(s, n);
    }
}
```

Hvilke af følgende kommandoer instantierer et objekt af typen OpgB?

- OpgB b = new OpgB(13);
- OpgA a = (OpgB) new OpgA(13);
- OpgB b = new OpgB("Noget", "Noget".length());
- OpgB b = new OpgB("Navn", "4");
- OpgB b = new OpgB();
- OpgB b = new OpgB(null, null);
- OpgB b = new OpgB(null, 0);

Bord nr.	Kursus nr.: 02101	Dato: 14. december 2018	Ark nr.
	Kursusnavn: Indledende Programmering		
	Studienr.: _____	Fødselsdato: _____	
	Navn: _____		

10. Betragt følgende stykker kode:

```
public abstract class E implements Comparable<E> {
    public final String name;
    protected long number;
    public E (String name, long l) {
        this.name = name;
        this.number = l;
    }
    public abstract String combineInfo();
    public long getNumber() {
        return number;
    }
}
```

Man vil gerne færdiggøre klassen F,

```
public class F extends E {
    // TODO
}
```

så den senere kan bruges sådan her:

```
F f = new F("F", (long) 0);
System.out.println(f.combineInfo());
```

Hvilke metoder er det *nødvendigt* at implementere i F, dér hvor der nu står “// TODO”, for at programmet virker?

- Metoden public String getString()
- Metoden public long getNumber()
- Metoden public int compareTo(F f)
- Metoden public int compareTo(E e)
- Metoden public static boolean compare(F f1, F f2)
- Metoden public String combineInfo()
- Konstruktøren F(String name, long l)
- Konstruktøren Comparable<E>(String name, long l, E e)

Bord
nr.

Kursus nr.: 02101

Dato: 14. december 2018

Ark nr.

Kursusnavn: Indledende Programmering

Studienr.: _____ Fødselsdato: _____

Navn: _____

Opgave 2 (Evaluering af udtryk) 8%

Skriv svaret på den efterfølgende linie.

1. Evaluer følgende udtryk, som de evalueres af Java:

$10 \% 2 \% 5 + 1$ _____

$10 / 2 / 5 + 1$ _____

$3 * 2 \% 4 + 7 \% 5$ _____

2. Evaluer følgende udtryk, som de evalueres af Java:

$10 \% 3 * 3 + "5"$ _____

$4 + 3 + "5" + 4 + 3$ _____

$(3*2 > 15 \% 10) == (4 > 8 \% 6 - 2)$ _____

Bord
nr.

Kursus nr.: 02101

Dato: 14. december 2018

Ark nr.

Kursusnavn: Indledende Programmering

Studienr.: _____ Fødselsdato: _____

Navn: _____

Opgave 3 (Løkker og metodekald) 22%

1. Betragt følgende stykke kode:

```
int i = 0;
for (int a = x; a >= 0; a--) {
    for (int b = -a; b <= a; b++) {
        System.out.print(a);
        i++;
    }
}
int j = -i;
```

Hvad skrives til `System.out` for følgende værdier af `x`?

`x = 2` _____

`x = 3` _____

`x = 4` _____

Hvor mange gange skrives tallet 5 til `System.out` for følgende værdier af `x`?

`x = 3` _____

`x = 5` _____

`x = 14` _____

Hvilken værdi bliver `j` tildelt for følgende værdier af `x`?

`x = 3` _____

`x = 4` _____

`x = 9` _____

`x = 19` _____

`x = -16` _____

Bord
nr.

Kursus nr.: 02101

Dato: 14. december 2018

Ark nr.

Kursusnavn: Indledende Programmering

Studienr.: _____ Fødselsdato: _____

Navn: _____

2. Betragt følgende metoder:

```
public static int delA(int i) {  
    return delB(i,1);  
}  
public static int delB(int i, int j) {  
    if(i<2) return j;  
    if(i%2 == 0) {  
        return delB(i/2,j+1);  
    } else {  
        return delB(i-1,j);  
    }  
}
```

Hvad returnerer delA ved følgende værdier for parameteren i?

i = 1 _____

i = 2 _____

i = 3 _____

i = 4 _____

i = 8 _____

Når man kalder delA bliver delB kaldt mindst 1 gang.

Hvor mange gange i alt bliver delB kaldt ved de følgende kald til delA?

delA(2) _____

delA(7) _____

delA(8) _____

delA(32) _____

Bord
nr.

Kursus nr.: 02101

Dato: 14. december 2018

Ark nr.

Kursusnavn: Indledende Programmering

Studienr.: _____ Fødselsdato: _____

Navn: _____

3. Betragt følgende stykke kode:

```
Scanner input = new Scanner(System.in);
int[] tabel = new int[input.nextInt()];
for (int i = tabel.length; i > 0; i--) {
    tabel[i-1] = input.nextInt();
}
System.out.println(Arrays.toString(tabel));
```

Hvad skrives til `System.out` hvis brugeren under programudførelsen indtaster følgende?

5 10 8 6 4 2 _____

2 10 11 _____

3 19 -4 0 _____

Er programmet robust overfor forskellig brugeradfærd?

Programmet er robust overfor alle input fra brugeren.

Programmet er ikke robust overfor følgende input,

da det ikke kun består af tal: _____

Programmet er robust overfor alle sekvenser af tal.

Programmet er ikke robust overfor følgende sekvens af tal:

Bord
nr.

Kursus nr.: 02101

Dato: 14. december 2018

Ark nr.

Kursusnavn: Indledende Programmering

Studienr.: _____ Fødselsdato: _____

Navn: _____

Opgave 4 (Forstå og forandre) 24%

1. Betragt følgende metode:

```
public static int opg1(int x) {  
    if (x <= 1) {  
        return x;  
    } else {  
        return x*opg1(x-1);  
    }  
}
```

Hvad returnerer metoden ved følgende værdier for parametren x:

x = 2 _____

x = -10 _____

x = 3 _____

x = 5 _____

Hvilket eller hvilke hvis nogen af følgende udsagn er korrekt(e) om opg1 ovenfor?

opg1 er rekursiv

opg1 er iterativ

Lav en metode med samme navn og opførsel som opg1, som benytter en for-løkke, som ikke benytter while-løkker, og som ikke kalder sig selv:

```
1 public static int opg1(int x){  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19 }
```

Bord
nr.

Kursus nr.: 02101

Dato: 14. december 2018

Ark nr.

Kursusnavn: Indledende Programmering

Studienr.: _____ Fødselsdato: _____

Navn: _____

2. Betragt følgende metoder:

```
public static void opg2(int i){
    System.out.print(" ");
    if(i % 2 == 0) {
        opg2a(i);
    } else {
        opg2b(i);
    }
}
```

```
private static void opg2a(int i) {
    System.out.print(i);
    opg2(i/2);
}
```

```
private static void opg2b(int i) {
    System.out.print(i);
    if(i==1) return;
    opg2(3*i+1);
}
```

Hvad skrives til `System.out` ved følgende metodekald?

`opg2(4)` _____

`opg2(10)` _____

`opg2(6)` _____

Hvor mange gange bliver `opg2b` kaldt ved følgende kald til `opg2`?

`opg2(4)` _____

`opg2(10)` _____

`opg2(3)` _____

`opg2(8)` _____

`opg2(64)` _____

**Bord
nr.**

Kursus nr.: 02101

Dato: 14. december 2018

Ark nr.

Kursusnavn: Indledende Programmering

Studienr.: _____ **Fødselsdato:** _____

Navn: _____

Lav en metode med samme navn og opførsel som `opg2`, som benytter en `while`-løkke, som ikke benytter `for`-løkker, og som ikke kalder sig selv:

```
1 public static void opg2(int x){
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26 }
```

Bord
nr.

Kursus nr.: 02101

Dato: 14. december 2018

Ark nr.

Kursusnavn: Indledende Programmering

Studienr.: _____ Fødselsdato: _____

Navn: _____

3. Betragt følgende kode:

```
public static int opg3(int l, int r) {
    if(r < l) return -opg3(r, l);
    int c = 0;
    for(int i = l; i<r; i++) {
        c++;
    }
    return c;
}
```

Hvad returneres ved følgende metodekald?

opg3(6,20) _____

opg3(20,6) _____

opg3(-10,10) _____

Skriv en metode med samme opførsel som opg3, men som hverken benytter løkker eller rekursive kald:

```
1 private static int opg3(int l, int r){
2
3
4
5
6
7
8
9
10
11
12
13 }
```

Bord nr.	Kursus nr.: 02101	Dato: 14. december 2018	Ark nr.
	Kursusnavn: Indledende Programmering		
	Studienr.: _____	Fødselsdato: _____	
	Navn: _____		

Opgave 5 (Færdiggør koden) 12%

På vejen Virtuelvej ligger husene med de lige husnumre på den ene side, og husene med de ulige husnumre på den anden side. På sin rute besøger postbuddet først de lige husnumre i stigende rækkefølge: 2, 4, 6, ... og dernæst de ulige husnumre i faldende rækkefølge: ..., 5, 3, 1.

Udvid klassen Husnummer, så den implementerer Comparable-interface. Sammenligningen skal være sådan, at Husnummer A er mindre end Husnummer B, når Husnummer A bliver besøgt før Husnummer B på ruten. For eksempel:

Husnr 16 < Husnr 7 Husnr 2 < Husnr 8 Husnr 5 < Husnr 3.

Metoden compareTo skal returnere -1, 0, eller 1.

-1 betyder `this < h`, 1 betyder `this > h`, og 0 betyder at de er lige store.

```

1  public class Husnummer implements Comparable<Husnummer>{
2      public final int nummer;
3      public Husnummer(int nummer) {
4          this.nummer = nummer;
5      }
6      public int compareTo(Husnummer h) {
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30  }
31 }
```


Bord nr.	Kursus nr.: 02101	Dato: 14. december 2018	Ark nr.
	Kursusnavn: Indledende Programmering		
	Studienr.: _____	Fødselsdato: _____	
	Navn: _____		

Opgave 6 (Find og ret fejl og færdiggør) 10%

Betragt følgende abstrakte klasse:

```
public abstract class Figur {
    protected String type;
    public abstract int getVolume();
    public String toString() {
        return type+" på "+getVolume()+" kubikcentimeter";
    }
}
```

På næste side findes en implementation af *Kasse* som har fejl. Find fejlene og marker dem ved at overstrege dem. Skriv dine egne rettelser i kolonnen til højre. Du behøver kun at notere, hvad den overstregte tekst skal erstattes af. Ved pladsmangel kan du notere linienumre og rettelser på et vedlagt papir.

- En kasse er en tredimensionel figur hvis volumen kan udregnes ved at gange dets højde, bredde og længde.
- En kube er en kasse, hvor højde, bredde og længde er ens.

Ret implementationen af *Kasse* og færdiggør implementationen af *Kube*, så `getVolume` og `type` er korrekte, således at f.eks. følgende kode

```
Kasse k = new Kasse(2,3,4);
System.out.println(k);

Kube t = new Kube(3);
System.out.println(t);
```

skriver følgende til `System.out`:

```
Kasse på 24 kubikcentimeter
Kube på 27 kubikcentimeter
```

Brug nedarvning til at undgå redundans i koden.

Klassen *Figur* øverst på denne side skal forblive uændret.

Bord
nr.

Kursus nr.: 02101

Dato: 14. december 2018

Ark nr.

Kursusnavn: Indledende Programmering

Studienr.: _____

Fødselsdato: _____

Navn: _____

Kasse:

```
1 public class Kasse implements Figur {
2     private int length;
3     private int width;
4     private int height;
5     private String type;
6
7     public Kasse(int l, int w, int h) {
8         l = length;
9         w = width;
10        h = height;
11        super.setType("Kasse");
12
13    }
14
15    public abstract int getVolume();
16
17
18
19
20
21
22
23 }
```

Kube:

```
1 public class Kube
2
3
4
5     public Kube (int s) {
6
7
8
9
10
11
12
13
14
15
16
17
18 }
```