1	1.1
	Hvilke (en eller flere) av følgende utsagn er korrekt? Velg ett eller flere alternativer
	■ En typeløs peker (<i>void peker</i>) er det nærmeste C++ kommer en ren maskinadresse
	Hvis man trenger kode som kjøres raskt er C++ et dårlig verktøy
	■ Templates kan ikke brukes når man lager generelle klasser
	C++ støtter objektorientert programmering
	Maks poeng: 5
2	1.2
	Hvilke (en eller flere) av følgende utsagn er korrekt? Velg ett eller flere alternativer
	☐ En klasse kan ha flere konstruktører
	■ Medlemmer i en klasse er public som standard (default)

☐ En klasse kan ikke ha flere destruktører

auto-typen er spesielt nyttig i generisk kode

³ 1.3

Hvilke (en eller flere) av følgende utsagn er korrekt?

Velg ett eller flere alternativer

this-pekeren til et objekt er lik pekeren til objektet

Det er ikke mulig å endre på this-pekeren i en medlemsfunksjon

En this-peker kan ikke legges til i en vektor

this-pekeren må slettes i destruktøren

4 1.4 - is Yellow

```
Kodesnutten under inneholder en eller flere feil.
```

```
enum class Color{Red = 0, Blue = 1, Yellow = 2};
bool isYellow(Color 2) {
    return c == 2;
}
Hvilke (en eller flere) av alternativene under fungerer som ønsket?
Velg ett eller flere alternativer
   enum class Color{Red = 0, Blue = 1, Yellow = 2}:
bool isYellow(Color 2) {
       return c == Color::Yellow;
   }
   enum class Color{Red = 0, Blue = 1, Yellow = 2};
bool isYellow(Color c) {
       return c == Yellow:
   }
   enum class Color{Red = 0, Blue = 1, Yellow = 2};
 bool isYellow(Color c) {
       return c == Color::Yellow;
   }
   enum class Color{Red = 0, Blue = 1, Yellow = 2};
bool isYellow(Color c) {
       return c == 2;
   }
```

⁵ 1.4 - diceValues

Kodesnutten under handler om terninger (*dice*), men den inneholder en eller flere feil som gjør at koden ikke vil kjøre/ikke vil kjøre som ønsket dersom *diceValues()*-funksjonen kalles fra f.eks. *main.cpp*-filen.

```
//The code in the file DiceValues.h
2 #pragma once
  void diceValues(int numberOfDice);
  //The code in the file DiceValues.cpp
1
2 #include "DiceValues.h"
   #include "std_lib_facilities.h"
4
   vector<int> diceValues(int numberOfDice) {
5
       vector<int> diceValues:
6
       constexpr int maxDiceValue = 6;
7
8
       for(int i = 0; i < numberOfDice; ++i) {</pre>
9
            int currValue = rand() % maxDiceValue;
10
            diceValues[i] = currValue;
11
       }
12
       return diceValues;
13
   }
14
```

På hvilke (velg en eller flere) kodelinjer oppstår feilen(e)? Bak hvert kodelinjetall-alternativ står det en parentes som indikerer om linjen er i *headerfilen* (.h) eller i *.cpp-filen* (.cpp).

Velg ett	eller	flere	altern	ativer
----------	-------	-------	--------	--------

7 (.cpp)

■ 13 (.cpp)

■ 11 (.cpp)

6 (.cpp)

1 (.cpp)

2 (.h)

3 (.h)

9 (.cpp)

■ 10 (.cpp)

⁶ 1.5

Se koden under.

```
#include "std_lib_facilities.h"
2
   class MyStack {
3
   public:
        vector<int> vec:
5
6
        void push(int number);
7
8
        void pop();
9
   };
10
11
   inline void MyStack::push(int number) {
12
        vec.push_back(number);
13
   }
14
15
   inline void MyStack::pop() {
16
        vec.pop_back();
17
   }
18
```

Vi ønsker å gjøre stakken om fra noe som bare håndterer heltall til noe som håndterer ulike typer ved å bruke templates. Bruk nedtrekksmenyen for å velge hvordan koden da skal se ut. Blank (tomt felt i nedtrekksmenyen) er også mulig. **Men husk å aktivt velge blankt felt i** nedtrekksmenyen dersom du mener det er rett svar på noen av plassene!

inline-nøkkelordet er brukt for å ha alt i headerfilen.

```
#include "std lib facilities.h"
```

```
Velg alternativ (template<T>, template<const T>, template<static T>, , template<typename T>)
```

```
class MyStack
                 Velg alternativ
                                  (<static T>, <typename T>, <const T>, , <T>){
public:
                           (<typename T>, <T>, , <static T>, <int>, <const T>) vec;
           Velg alternativ
               Velg alternativ
                               (, template T, static T, const T, int, T) number);
  void push(
  void pop();
};
  Velg alternativ
                  (template<typename T>, template<T>, template<static T>, , template<const
T>)
                                      (<int>, <typename T>, , <const T>, <static T>, <T>)::push(
inline void MyStack
                     Velg alternativ
                  (const T, typename T, int, T, static T, ) number) {
  Velg alternativ
                    Velg alternativ
                                     (const T, int, T, static T, , typename T)number);
  vec.push back(
}
  Velg alternativ
                  (template<static T>, , template<typename T>, template<T>, template<const
T>)
                                      (<static T>, , <typename T>, <int>, <T>, <const T>)::pop() {
inline void MyStack Velg alternativ
  vec.pop_back();
}
```

⁷ 1.6

Se på koden under.

```
#include "std_lib_facilities.h"
 class Cat {
 protected:
     string name;
     int age;
 public:
     Cat(string name, int age):
        name{name}, age{age} {}
     virtual void meow() {cout << "Meow\n";}</pre>
     Cat& operator=(const Cat&) = delete;
      ~Cat() {}
 };
 class NorwegianCat : public Cat {
 public:
     NorwegianCat(string name, int age = 12) :
        Cat{name, age} {}
     void meow() override {cout << "Meow (in Norwegian)\n";}</pre>
 };
Hvilke (en eller flere) av de følge påstandene er korrekte?
Velg ett eller flere alternativer
 NorwegianCat har tilgang til medlemsvariablene i Cat-klassen
 Funksjonen meow() i Cat er virtual
 Funksjonen meow() i NorwegianCat er virtual
 Cat er en abstrakt klasse
```

8	1	7

Hvilke (en eller flere) av alternativene er nødvendige når det brukes dynamisk minne i en klass	se?
Velg ett eller flere alternativer	

Konstruktør

Destruktør

☐ Operatoroverlasting av *operator*=

Grunn kopiering

⁹ 1.8

Gitt klassedeklarasjonen for Hometown under.

```
class Hometown{
    string name;
public:
    Hometown(string name) : name{name} {}
};
```

I hvilke (en eller flere) av klassedeklarasjonene under kan det være lurt å implementere kopikonstruktøren selv?

Velg ett eller flere alternativer

```
class Person {
      string name;
      Hometown * home = nullptr;
public:
      Person(const Person & Me) = default;
      Person(string name, string city) : name{name} {home = new Hometown(city);}
  };
  class Person {
       string name;
       Hometown home;
public:
       Person(string name, string city) : name{name}, home{city} {}
       Person& operator=(Person&) = delete;
  };
   class Person {
       string name;
       Hometown home;
public:
       Person(string name, string city) : name{name}, home{city} {}
       Person() = delete;
  };
   class Person {
      string name;
      Hometown * home = nullptr;
      Person(string name, string city) : name{name} {home = new Hometown(city);}
   };
```

¹⁰ 1.9

Gitt følgende kodesnutt:

```
#include "std_lib_facilities.h"

int main(){
    vector<int> myVec{2, 5, 42, 8};
    auto iter = myVec.begin();
    vector<int>* ptr = &myVec;
    return 0;
}

Hvilke (en eller flere) av disse påstandene stemmer?

velg ett eller flere alternativer

ptr og iter peker på det samme

(*ptr).at(2) og ptr->at(2) betyr det samme
```

Maks poeng: 5

iter[3] og *(myVec.end()-1) betyr det samme

iter og myVec.rend() peker på samme element

¹¹ 2.1 - konstante referanser

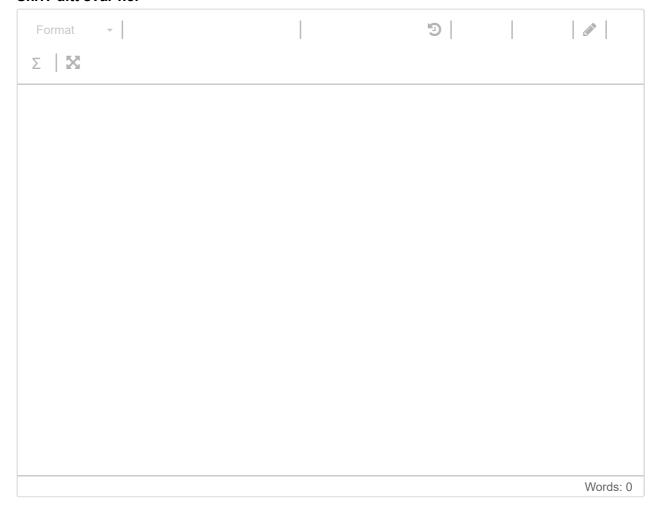
 $\label{thm:constraint} \mbox{Hvilke fordeler er det med konstante referanser ($\it const \, reference$)? \mbox{\bf Forklar kort.}$

Skriv ditt svar her Format Σ S S Words: 0

¹² 2.1 - smartpekere

Hvilke fordeler er det med smartpekere framfor "vanlige" pekere? Forklar kort.

Skriv ditt svar her



¹³ 2.2

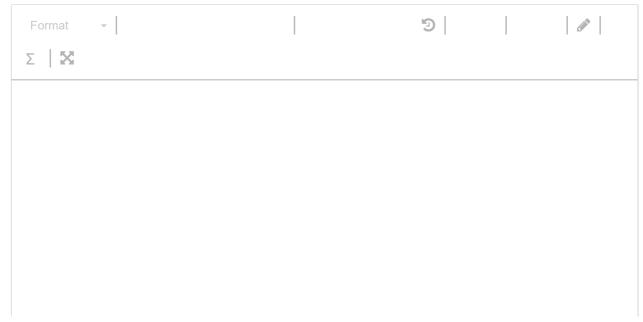
Vær oppmerksom på at denne oppgaven har to deloppgaver (a og b). Spesifiser hvilken deloppgave du svarer på ved å først skrive **a)** eller **b)**.

Se på klassedeklarasjonene under:

```
class Node {
       int nodeValue;
       Node * nextNode = nullptr;
4
   public:
5
       Node(int nodeValue) : nodeValue{nodeValue} {}
       void addNode(int nodeValue) {
            nextNode = new Node(nodeValue);
       }
10
       Node * getNextNode(){return nextNode;}
11
       int returnValue(){return nodeValue;}
12
13
       ~Node() {
            if(nextNode) delete nextNode;
15
       }
16
   };
17
```

- a) Hvorfor risikerer koden minnelekkasje?
- b) Hvordan kan problemet løses?

Skriv ditt svar her



5/1/23, 9:5	57 AM	TDT4102 Insperaøving 2 - 25.04.23 (en/nb/nn)	
		,	Words: 0

¹⁴ 2.3 - Person-struct

Vær oppmerksom på at denne oppgaven har to deloppgaver (a og b). Spesifiser hvilken deloppgave du svarer på ved å først skrive **a)** eller **b)**.

Se på klassedeklarasjonene under:

```
#include <iostream>
   using namespace std;
3
   struct Person {
   private:
      string name;
6
      int age;
7
      string address;
8
9
   public:
10
     Person(string name, int age, string address)
        : name{name}, age{age}, address{address} {}
12
      string getName() { return name; }
13
      const int getAge() { return age; }
14
      string getAddress() const { return address; }
15
   };
16
17
   int main() {
18
      const Person ola{"Ola", 50, "ola@nordmann.no"};
19
      cout << "Name: " << ola.getName() << endl;</pre>
20
      cout << "Age: " << ola.getAge() << endl;</pre>
21
      cout << "Email: " << ola.getAddress() << endl;</pre>
22
     return 0;
   }
```

- a) Hva er problemet med koden? Forklar kort.
- **b)** Hvilke (en eller flere) endringer må til i klassedeklarasjonen for at koden i *main()* skal kjøre som forventet? **Forklar kort.** Skriv forslag til kode som løser problemet, men kun hva du vil legge til/fjerne/endre og hvor i koden dette er. <u>Du trenger ikke å skrive inn hele koden på nytt.</u>

Skriv ditt svar her



5/1/23, 9:5	57 AM TDT4102 Insperaøving 2 - 25.04.23 (en/nb/nn)		
		Words: 0	+
		TVOIGO. U	

¹⁵ 2.3 - Banana

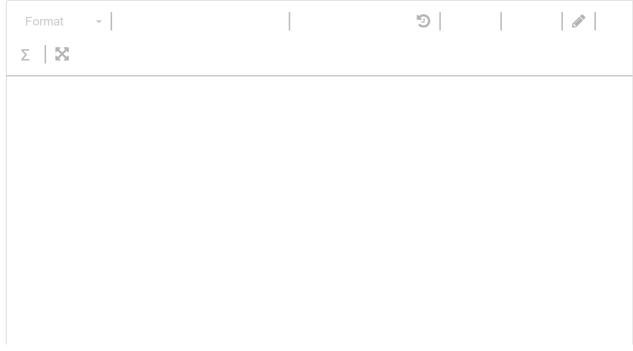
Vær oppmerksom på at denne oppgaven har to deloppgaver (a og b). Spesifiser hvilken deloppgave du svarer på ved å først skrive **a)** eller **b)**.

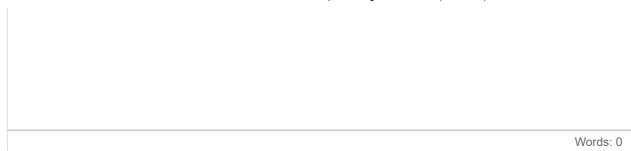
Se på klassen under:

```
//This code is in the file Banana.h
   #pragma once
   #include "std_lib_facilities.h"
  class Banana {
       int sweetness = 70;
       int weight = 100;
       bool isRipe = true;
   public:
9
       friend ostream& operator << (ostream& os, const Banana & b);
10
   };
11
  //This code is in the file Banana.cpp
  #include "Banana.h"
   ostream& Banana::operator<<(ostream& os, const Banana & b){
           os << b.weight << " grams banana with " << b.sweetness << ".";
           if(b.isRipe) os << " Ready to be eaten!";</pre>
           return os;
   }
```

- a) Hva er problemet med klassen? Forklar kort.
- **b)** Hvordan kan problemet med denne koden minimum løses? (Med andre ord: Hva er den enkleste måten å løse dette problemet på?) **Forklar kort.** Skriv forslag til kode som løser problemet, men kun hva du vil legge til/fjerne/endre og hvor i koden dette er. <u>Du trenger ikke å skrive inn hele koden på nytt.</u>

Skriv ditt svar her





¹⁶ 2.4 - .at()

Hvorfor er det lurt å bruke vektorfunksjonen .at() i stedet for tabellindeksoperatoren (operator[]) dersom man vil aksessere et element i en vektor? **Forklar kort.**

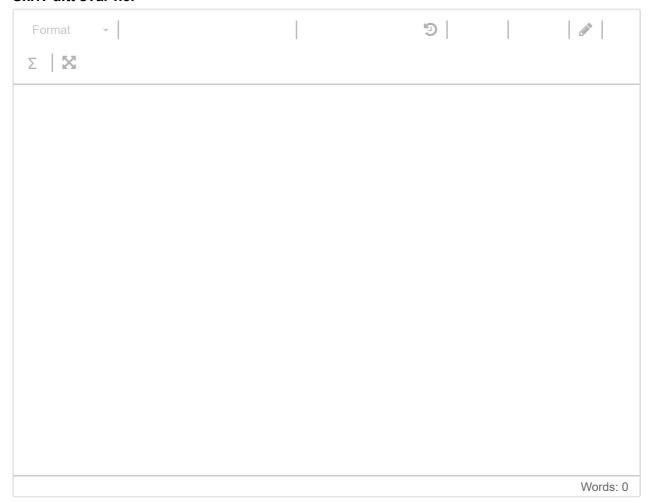
Skriv ditt svar her



¹⁷ 2.4 - tilordningsoperator

Hvorfor kan det være nyttig å returnere en referanse til et objekt i implementasjon av tilordningsoperatoren? **Forklar kort.**

Skriv ditt svar her



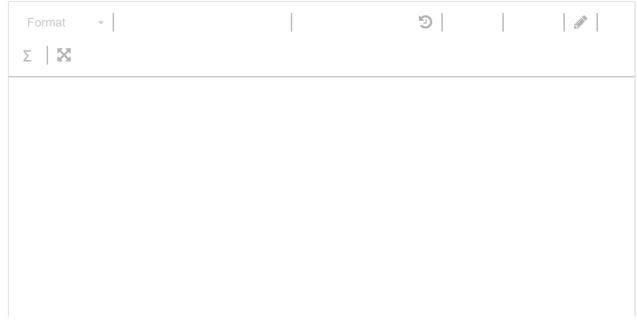
¹⁸ 2.5

Se på koden under:

```
#include "std_lib_facilities.h"
   class Drink{
   public:
        virtual void serveDrink() const {cout << "Here is your drink!\n";}</pre>
   };
   class Tea : public Drink{
   public:
        void serveDrink() const override {cout << "Here is your tea!\n";}</pre>
   };
10
   class HotChocolate : public Drink{
11
   public:
        void serveDrink() const override {cout << "Here is your hot chocolate!\n";}</pre>
13
   };
14
15
   int main(){
16
        vector<Drink*> drinks;
17
18
        Tea earlGrey{};
19
        HotChocolate cocoa{};
        Drink mysteryDrink{};
21
        Tea greenTea;
23
        drinks.push_back(&earlGrey);
24
        drinks.push_back(&cocoa);
        drinks.push_back(&mysteryDrink);
26
        drinks.push_back(&greenTea);
28
        for(auto d : drinks){
29
            d->serveDrink();
31
   }
```

Hvorfor lønner det seg å la elementene i vektoren *drinks* i *main()* være av typen *Drink** isteden for typen *Drink?* **Forklar kort.**

Skriv ditt svar her



5/1/23, 9:5	7 AM	TDT4102 Insperaøving 2 - 25.04.23 (en/nb/nn)	
			Words: 0

¹⁹ Nedlasting/opplasting av kode v23.1

LAST NED UTDELT KODE

handout 2023.zip

Se instruksjoner på forrige side

LAST OPP

Last opp all den komplette koden som en .zip-fil. Ikke endre den opprinnelige mappestrukturen. For å få bestå prøven er det **HELT AVGJØRENDE AT DU LASTER OPP DEN NYE ZIP-FILEN DIN KORREKT**, minst en gang i løpet av de 4 timene du har til rådighet.

Det er mulig å oppdatere både enkeltsvarene og filopplastningen **flere ganger**, i tilfelle du retter på noe etter første innlevering.

Prøv å last opp en oppdatert zip-fil minst en gang ekstra, midt i prøvetiden, for å se at du klarer det, og for å finne ut hvor lang tid du bruker på det.

Last opp zip-filen med besvarelsen din her. Alt i én zip-fil.

