NTNU

Kandidat nr./*Candidate no.* 10159

Dato/*Date:* 10.6.2016 Side/*Page:* 1

Emnekode/*Subject* TDT4102 / pros. og. obj. prog.

Antall ark/*Number of pages:* 13

Denne kolonnen er forbeholdt sensor

This column is for external examiner

Oppgave 1

1a) 2
3
4

1b) First i = 8
Second i = 4

1c) 28, 0, 5, 20

1d) 20 30

1e) Instanser : 8
Destruktør : 1

1f) (i) Animal : Garfield
Dog : Lassie
~~Animal : Lassie~~
Dog : Lassie

(ii) Animal : ~~Garfield~~
Dog : Lassie
~~Dog~~ : Lassie
Dog : Lassie

(iii) Animal : "navn" (hvis virtual)

1g) 3, Old Splitken

1h) 321 aaa

1i) danger value = 5

(pga short circuit evaluation)

Fagtrykk Trondheim AS 12 2015

Den gule kopien beholder du/*Please keep the yellow page*

NTNU

Kandidat nr./*Candidate no.* 10159

Dato/*Date:* 10.6.2016 Side/*Page:* 2

Emnekode/*Subject* TDT4102/prog. og obj. prog.

Antall ark/*Number of pages:* 13

Denne kolonnen er forbeholdt sensor

This column is for external examiner

1j) caught e = 3

Oppgave 2

```
2a) int checkAlarm(const double value, const double threshold)
    {
        if (value > threshold) && value <= MAX_WIND)
            return 1;
        else if (value < 0 || value > MAX_WIND)
            return -1;
        else
            return 0;
    }
```

```
2b)
void printAlarm(const int id, const double wind, const int dir)
{
    cout << "ALARM:" << id << ": " << wind
         << setprecision(2) << wind << "(fra: ";
    [illegible] dir [illegible]) cout << "North";
    else if([illegible] dir [illegible] <= 135) cout << "East";
    else if ([illegible] dir && dir < 225) cout << "South";
    else ~~if~~ cout << "West";

}
```

Den gule kopien beholder du/*Please keep the yellow page*

Fagtrykk Trondheim AS 12.2015

NTNU

Kandidat nr./*Candidate no.* 10159

Dato/*Date:* 10.6.2016 Side/*Page:* 3

Emnekode/*Subject* TDT4102 / pros og. obj. prog. Antall ark/*Number of pages:* 13

Denne kolonnen er forbeholdt sensor

This column is for external examiner

```
2c)  ostream& operator<<(ostream& os, const sample& s)
     {
          os << s.id << ' ' << s.wind
             << ' ' << s.dir; << '\n';
          return os;
     }

2d). void readMeasurements(const char* filename)
     {
          ifstream file(*filename);
          if (file.fail()){
               cerr << "File couldn't be opened";
               return;
          }
          sample temp = {0};
          sample temp2 = {0};
          int mill = 0; int smp = 0;
          while (file >> temp.id){
                 file >> temp.wind;
                 file >> temp.dir;
                 measurements[mill][smp] = temp;
                 smp = ++smp % N_SAMPLES;
                  if (smp == 0) ++mill;
          }
          file.close(); ++mill;
          File.close();
     }
```

Fagtrykk Trondheim AS 12.2015

NTNU

Kandidat nr./*Candidate no.* 10159

Dato/*Date:* 10.6.2016 Side/*Page:* 4

Emnekode/*Subject* TDT 4102 / pros. og. obj. prog. Antall ark/*Number of pages:* 13

Denne kolonnen er forbeholdt sensor

This column is for external examiner

2c)
```
vector<sample> calcStats()
{
        vector<sample> result;
        for (int mill = 0; mill < N_MILLS; ++mill) {
          int mill_id = measurements[mill][0].id;
          double mill_wind = avgWind (measurements[mill]) [illegible]
          int mill_dir = strongestDir (measurements[mill]);
          ~~result;~~
          sample temp = {mill_id, mill_wind, mill_dir};
          result.push_back(temp);
        }
        return result;
}
double avgWind(double* sampleArr)
{
        double sum = 0;
        for (int i = 0; i < N_SAMPLES; ++i){
            sampleArr[i]ple Arr[i].wind;
        }
        return sum / N_SAMPLES;
}
```

NTNU

Kandidat nr./*Candidate no.* 10159

Dato/*Date:* 10.6.2016 Side/*Page:* 5

Emnekode/*Subject* TDT4102 /pros. og obj. prog.

Antall ark/*Number of pages:* 13

Denne kolonnen er forbeholdt sensor

This column is for external examiner

```
int strongestDir (sample* sampleArr)
{
    ~~double bestWind = -1 sampleArr[0].wind;~~
    int bestDir = -1 sampleArr[0].dir;
    for (int i = 0; i < N_SAMPLES; ++i){
        if (sampleArr[i].wind > bestWind){
            bestWind = sampleArr[i].wind;
            bestDir = sampleArr[i].dir;
        }
    }
    return bestDir;
}

2f) bool operator<(const sample& lhs, const sample& rhs)
    {
        return lhs.wind < rhs.wind;
    }

2g) int checkMeasurements()
    {
        int ~~totalMissingCount~~ = 0;
        for (int mill = 0; mill < N_MILL; ++mill){
int millMissCount;
int millMissingCount;
            for (int smp = 0; smp < N_SAMPLES; ++smp){
                sample measurement = measurement[mill][smp];
                if (measurement.wind == -1 || )
                    ++millMissCount;
                    ++totalMissingCount;
            }
```

Den gule kopien beholder du/*Please keep the yellow page*

Fagtrykk Trondheim AS 12.2015

NTNU

Kandidat nr./*Candidate no.* 10159

Dato/*Date*: 10.6.2016 Side/*Page*: 6

Emnekode/*Subject* TDT4102 / prog. og obj. prog.

Antall ark/*Number of pages*: 13

Denne kolonnen er forbeholdt sensor

This column is for external examiner

```
                if (millMissCount > 1)
                    throw measurement.id;
            }
        }
        return totalMissingCount;
    }

2h) void repairMeasurements()
    {
        for (int mill = 0; mill < N_MILLS; ++mill){
            for (int smp=0; smp < N_SAMPLES; ++smp){
                sample* measurementPtr = &measurements[mill][smp];
                if (~~mill~~measurementPtr -> wind ~~!~~== -1.0)
                    continue;
                if (smp == 0)
                    measurementPtr -> wind
                        = measurements[mill][1];
                else if (smp == ~~N~~_SAMPLES - 1)
                    measurementPtr -> wind
                        = measurements[mill][~~S~~smp - 1];
                else {
                    measurementPtr -> wind
                        = (measurements[mill][smp - 1]
                            + measurements[mill][smp+1])
                            / 2;
                }
```

Den gule kopien beholder du/*Please keep the yellow page*

Fagtrykk Trondheim AS 12 2015

NTNU

Kandidat nr./Candidate no. 10K9

Dato/Date: 10.6.2016 Side/Page: 7

Emnekode/Subject TDT4102 / pros og obj prog Antall ark/Number of pages: 13

Denne kolonnen er forbeholdt sensor

This column is for external examiner

```
        }
     }
  }
```

Oppgave 3

3a)
```
class Person {
private:
        const int id;
        string email;
public:
        Person(const int id, const string& email);
        int getId() const;
        string getMail() const;
        void setMail(const string& email);
};
```

3b)
```
Person::Person(const int id, const string& email)
        : id(id), email(email)
{}
int Person::getId() const { return id; }
string Person::getMail() const { return email; }
void Person::setMail(const string& email){
        this->email = email;
```

Den gule kopien beholder du/*Please keep the yellow page*

Fagtrykk Trondheim AS 12.2015

Kandidat nr./*Candidate no.* 10159

Dato/*Date:* 10.6.2016 Side/*Page:* 8

Emnekode/*Subject* TDT4102 / pros. og obj. prog. Antall ark/*Number of pages:* 13

Denne kolonnen er forbeholdt sensor

This column is for external examiner

```
3c) class Driver : public Person{
    private:
        string carType;
        int    freeSeats;
    public:
        Driver(int id, string email, string carType
                , int freeSeats)
            : Person(id, email), carType(carType)
            , freeSeats(freeSeats)
            {}
    };

3d) Meeting::Meeting(int day, int start, int end, Campus
                     , Campus location, Person* owner)
        : day(day), start(start), end(end)
        , location(location), owner(owner)
        , driver(nullptr), firstPart(nullptr)
    {} , allMeet
     {
        allMeetings = this;
     }
```

Den gule kopien beholder du/*Please keep the yellow page*

Fagtrykk Trondheim AS 11 2012

NTNU

Kandidat nr./*Candidate no.* 10159

Dato/*Date*: 10.6.2016 Side/*Page*: 9

Emnekode/*Subject*: TDT4102 / pros. og. obj. prog

Antall ark/*Number of pages*: 13

Denne kolonnen er forbeholdt sensor

This column is for external examiner

```
3e) void Meeting::addParticipant(Person* person)
    {
        [illegible](firstPart == nullptr ) {
            firstPart = person;
            return;
        }
        Person* walker = firstPart;
        while (!walker->getNext() ) {
            walker = walker->getNext();
        }
        walker->setNext(person);
    }

3f) void Meeting::coDriving()
    {
        Meeting* walker = allMeetings;
        set<Meeting*>   otherMeetings;

        while (!walker->getNext() ){
            if (walker != this) && walker->day == day
              && walker->start == start && walker->end
                == end)
              ortherMeetings.insert(walker);

            walker = walker->next;
        }
```

Den gule kopien beholder du/*Please keep the yellow page*

NTNU

Kandidat nr./*Candidate no.* 10159

Dato/*Date:* 10.6.2016 Side/*Page:* 10

Emnekode/*Subject* TDT4102 / prog. og obj. prog Antall ark/*Number of pages:* 13

Denne kolonnen er forbeholdt sensor

This column is for external examiner

```
    cout << "Possible co-driving for meeting in"
         << location << " on " << day
         << " from " << start << " to "
         << end << " by <" <<
          owner->getMail() << ">" << endl;
    for (auto meetPtr : otherMeetings){
         cout << "\t Meeting by "<"
         << meetPtr->owner->getMail() << ">\n";
    }
}
3g) Meeting::~Meeting()
    {
        ~~Participant Par [illegible] nullptr){~~
        prev Participant ~~=~~ *prev, *current;
        w    prev = current = firstPart;
             while (current->getNext()){
                  current ~~=~~ current->getNext();
                  delete prev;
                  prev = current;
             }
             delete ~~prev~~current;
    }
}
```

Den gule kopien beholder du/*Please keep the yellow page*

Fagtrykk Trondheim AS 11 2012

NTNU

Kandidat nr./*Candidate no.* 10149

Dato/*Date:* 10.6.2016 Side/*Page:* 11

Emnekode/*Subject* TDT4102 / pros. og obj. prog. Antall ark/*Number of pages:* 13

Denne kolonnen er forbeholdt sensor

This column is for external examiner

Oppgave 4

```
4a) Dataset :: Dataset(int N)
        : N(N), y(new double[N])
    {
        for (int i = 0; i<N; ++i){
            y[i] = 0.0;
        }
    }

4b) Dataset :: Dataset(const Dataset& other)
    {  :  N(other.N), y(new double[N])
    {
        for (int i=0; i<N; ++i){
            y[i] =  other.y[i];
        }
    }

4c) Dataset :: ~Dataset ()
    {
        delete [] y;
    }
```

Den gule kopien beholder du/*Please keep the yellow page*

Fagtrykk Trondheim AS 11 2012

NTNU

Kandidat nr./*Candidate no.* 10159

Dato/*Date:* 10.6.2015 Side/*Page:* 12

Emnekode/*Subject* TDT4102/pros.og obj. Prog. Antall ark/*Number of pages:* 13

Denne kolonnen er forbeholdt sensor

This column is for external examiner

```
4d) Dataset& Dataset::operator=(Dataset rhs)
    {
          N = rhs.N;
          swap(y, other.y);
    };    return *this;
    }

4e) double Dataset::interpolate(double x)
    {
          ~~int (x[illegible]N)~~
                ~~return static_cast~~
          if (x >= N-1)
                return y[N-1];


          int x_down = static_cast<int>(x);
          int x_up   = x_down + 1;
          ~~int y_down =~~
          return   y[x_down] + (y[x_up] - y[x_down])
                      * (x - x_down);
          // x_{i+1} - x_i = x_i + 1 - x_i = 1 så unngår divisjon
    }
```

Fagtrykk Trondheim AS 11 2012

NTNU

Kandidat nr./*Candidate no.* 10159

Dato/*Date:* 10.6.2016 Side/*Page:* 13

Emnekode/*Subject* TDT4102/pros og obj. prog. Antall ark/*Number of pages:* 13

Denne kolonnen er forbeholdt sensor

This column is for external examiner

4 f)
```
Dataset* random_data(double min, double max, int size)
{
    Dataset* data = new Dataset(size);
    for (int i=0; i<size; ++i){
        (*data)[i] = random_double(min, max);
    }
    return data;
}

double randomDouble(double min, double max)
{
    double seed = static_cast/static_cast<double>(RAND_MA
    // tall mellom 0 og 1.

    return seed*(max-min) + min;
}
```

Den gule kopien beholder du/*Please keep the yellow page*

Fagtrykk Trondheim AS 11 2012