

# **TÖL-212M Rökstudd forritun**

## **Vikublað 1**

### **Reasoned Programming**

### **Weekly 1**

Snorri Agnarsson

10. janúar 2025

#### **Efnisyfirlit**

<b>1</b>	<b>Lýsing í kennsluskra — Course Description</b>	<b>2</b>
1.1	Námskeiðslýsing — Description: . . . . .	2
1.2	Hæfniviðmið — Learning Outcomes: . . . . .	2
<b>2</b>	<b>Gradescope og Piazza — Gradescope and Piazza</b>	<b>3</b>
<b>3</b>	<b>Áætlun — Plan</b>	<b>3</b>
<b>4</b>	<b>Plan</b>	<b>7</b>
<b>5</b>	<b>Vikuleg verkefni</b>	<b>10</b>
<b>6</b>	<b>Weekly Assignments</b>	<b>10</b>
<b>7</b>	<b>Lágmörk verkefnaskila</b>	<b>10</b>
<b>8</b>	<b>Minimum Assignment Counts</b>	<b>10</b>
<b>9</b>	<b>Einkunnagjöf</b>	<b>11</b>
<b>10</b>	<b>Grade Computations</b>	<b>11</b>
<b>11</b>	<b>Fyrirlestrar og æfingatímar</b>	<b>11</b>
<b>12</b>	<b>Lectures and Exercises</b>	<b>11</b>

<b>13 Almennt ítarefni</b>	<b>12</b>
<b>14 Reference Material</b>	<b>12</b>
<b>15 Efni vikunnar</b>	<b>13</b>
<b>16 This Weeks Topics</b>	<b>13</b>
<b>17 Verkefni</b>	<b>13</b>
<b>18 Assignments</b>	<b>14</b>

## 1 Lýsing í kennsluskrá — Course Description

Í kennsluskránni stendur eftirfarandi um námskeiðið.

In the course catalog the course is described as follows.

### 1.1 Námskeiðslýsing — Description:

Fjallað er um grundvallaratriði í rökstuddri forritun. Áhersla verður lögð á að nota rökstudda forritun til að þróa traust og sönnuð afbrigði af vel þekktum reikniritum, einkum á sviði leitar, röðunar og tvíleitartjáa. Meðal reiknirita sem fjallað verður um eru ýmis afbrigði af insertion sort, selection sort, quicksort, helmingunarleit og leit í tvíleitartré. Áhersla verður lögð á að dýpka skilning nemenda á reikniritunum ásamt því að ná góðum tókum á rökstuddri forritun. Að hluta verða verkefni leyst með hjálp sannreyingartóla svo sem Dafny eða OpenJML.

Fundamental concepts in program verification and reasoned programming are covered. Emphasis is placed on using reasoned programming to develop solid and proved versions of well-known algorithms, especially in the areas of searching, sorting, and binary search trees. Among algorithms covered are various versions of insertion sort, selection sort, quicksort, binary search, and searching in binary search trees. Emphasis will be placed on deepening the understanding of the algorithms as well as getting a good grip on program verification. Some exercises will use verification tools such as Dafny or OpenJML.

### 1.2 Hæfniviðmið — Learning Outcomes:

Að námskeiði loknu á nemandi að

- skilja vel grunnhugtök rökstuddrar forritunar svo sem Hoare þrenndir, fastayrðingar lykkju, fastayrðingar gagna, forskilyrði og eftirskilyrði,
- geta beitt rökstuddri forritun til að leysa algeng hagnýt vandamál á þann hátt að lausnin sé sönnuð,
- geta notað sannreyingartól svo sem Dafny og OpenJML til að sannreyna einföld reiknirit.

After completing this course the student should

- have a good understanding of the fundamental concepts of program verification such as Hoare triples, loop invariants, data invariants, preconditions, and postconditions,
- be able to use reasoned programming to solve common practical programs in a verified manner,
- be able to use verification tools such as Dafny and OpenJML to verify simple algorithms.

## 2 Gradescope og Piazza — Gradescope and Piazza

Við munum nota Piazza fyrir samskipti okkar á milli þegar það hentar. Þeir sem voru skráðir í námskeiðið í byrjun janúar eru þegar skráðir í námskeiðið í Piazza. Aðrir nemendur geta skráð sig sjálfir gegnum hlekkinn <https://piazza.com/hi.is/spring2025/tl212m>.

Gradescope<sup>1</sup> verður notað fyrir verkefnaskil. Þeir nemendur sem skráðir voru í námskeiðið í byrjun misseris með löglegt tölvupóstfang eru þegar skráðir í námskeiðið í Gradescope. Aðrir geta skráð sig sjálfir með því að nota Entry Code 6J3PP6.

We will use Piazza to communicate among ourselves when useful Those who were registered for the course at the beginning of January are already registered for the course in Piazza. Others can register themselves using the link <https://piazza.com/hi.is/spring2025/tl212m>.

Gradescope<sup>2</sup> will be used for turning in assignments. Those students who were registered for the course at the beginning of the semester with a valid email address are already registered for the course in Gradescope. Others can register themselves using Entry Code 6J3PP6.

## 3 Áætlun — Plan

Kennsluáætlunin er í Canvas í skránni `plan.pdf`. Eftirfarandi listi af efnisatriðum telur upp efni sem við höfum hug á að fjalla um, en vegna þess að þetta námskeið hefur ekki mjög oft verið kennt áður og ekki er komin mikil reynsla á hraða umfjöllunar er mögulegt að listinn eftir því sem líður á misserið. Það er kannski ólíklegt að unnt verði að fjalla um allt í listanum, en það mun koma í ljós. Listinn er ekki endilega í tímaröð.

- Stöðulýsingar.
- Hoare þrenndir:  $\{F\}S\{E\}$ .
  - Forskilyrði forritskafla.
  - Eftirskilyrði forritskafla.
- Notkunarlýsingar falla.
  - Forskilyrði falla.

---

<sup>1</sup><https://gradescope.com>

<sup>2</sup><https://gradescope.com>

- Eftirskilyrði falla.
- Fastayrðingar lykkju.
- Dafny.
  - Forskilyrði og eftirskilyrði.
    - \* requires
    - \* ensures
  - Fastayrðingar lykkju.
    - \* invariant
- Margföldun jákvæðra heiltalna.
- Veldishafning.
- Ýmsar summuformúlur og sönnun þeirra.
- Fylki og runur (arrays and sequences).
- Línuleg leit (linear search).
- Helmingunarleit (binary search) í fylki.
  - Í vaxandi (ascending) fylki
    - \* Finna eitthvað  $= x$
    - \* Finna fremsta  $\geq x$
    - \* Finna aftasta  $\leq x$
    - \* Finna fremsta  $> x$
    - \* Finna aftasta  $< x$
    - \* Finna fjölda  $= x$
    - \* Finna fjölda  $< x$
    - \* Finna fjölda  $\leq x$
    - \* Finna fjölda  $\geq x$
    - \* Finna fjölda  $> x$
    - \* Með fylki af fastri stærð, án lykkju
  - Í minnkandi (descending) fylki
    - \* Finna eitthvað  $= x$
    - \* Finna fremsta  $\leq x$
    - \* Finna aftasta  $\geq x$
    - \* Finna fremsta  $< x$

- \* Finna fremsta  $> x$
- \* Finna fjölda  $= x$
- \* Finna fjölda  $> x$
- \* Finna fjölda  $\geq x$
- \* Finna fjölda  $\leq x$
- \* Finna fjölda  $< x$
- Helmingunarleit með endurkvæmni.
- Helmingunarleit með lykkju.
- Helmingunarleit að rót samfellds falls.
  - Með lykkju.
  - Með endurkvæmni.
- Helmingunarleit að staðbundnu lággildi samfellds falls.
  - Með lykkju.
  - Með endurkvæmni.
- Mengi og pokar (sets, multisets, bags).
- Röðun, skilgreining með runum og pokum.
- Selection sort.
- Insertion sort.
- Quicksort.
  - Lomuto quicksort.
  - Dijkstra quicksort.
  - Hoare quicksort.
  - Yaroslavskiy quicksort (dual-pivot).
  - Kushagra quicksort (triple-pivot).
  - Og kannski önnur afbrigði.
- Mergesort.
- Fastayrðingar gagna.
- Tvíleitartré.
  - Tvíleitarskilyrði.
  - Tvíleitartré sem mengi.

- Tvíleitartré sem pokar.
- Fastayrðing gagna eða forskilyrði/eftirskilyrði.
- Leit, með lykkju og með endurkvæmni.
  - \* Eitthvað  $= x$
  - \* Fremsta  $= x$
  - \* Aftasta  $= x$
  - \* Fremsta  $\geq x$
  - \* Aftasta  $\leq x$
  - \* Fremsta  $> x$
  - \* Aftasta  $< x$
- Innsetning.
- Eyðing.
- AVL tré.
  - \* AVL skilyrði.
  - \* Fastayrðing gagna eða forskilyrði/eftirskilyrði.
- AA tré.
- Rauð-svört tré.
- 2-3 tré.
- Eintengdir listar.
  - Viðsnúningur lista.
  - Samskeyting lista.
  - Síun lista.
  - Innsetning í raðaðan lista.
  - Eintengdir listar í Dafny.
  - Mergesort fyrir eintengda lista.
  - Quicksort fyrir eintengda lista.
- Samplesort.
- Multiway mergesort.
- Parallel samplesort.
  - Ásamt margfaldri superscalar helmingunarleit.
- Parallel multiway mergesort.

## 4 Plan

The plan for the semester is in Canvas in the file `plan.pdf`. The following list contains topics we are interested in covering, but because this course has not very often been taught before we still do not have extensive experience about the speed of coverage so the list might change as the semester progresses. It is unlikely we will cover everything in the list and the list is not in time order.

- State descriptions.
- Hoare triples:  $\{F\}S\{E\}$ .
  - Preconditions of code segments.
  - Postconditions of code segments.
- Usage descriptions for functions.
  - Preconditions of functions.
  - Postconditions of functions.
- Loop invariants.
- Dafny.
  - Pre- and postconditions.
    - \* requires
    - \* ensures
  - Loop invariants.
    - \* invariant
- Multiplication of positive integers.
- Raising a number to a power.
- Various sum formulas and their proofs.
- Arrays and sequences.
- Linear search.
- Binary search for arrays.
  - In ascending array
    - \* Find some  $= x$
    - \* Find first  $\geq x$
    - \* Find last  $\leq x$
    - \* Find first  $> x$

- \* Find last  $< x$
- \* Find count of  $= x$
- \* Find count of  $< x$
- \* Find count of  $\leq x$
- \* Find count of  $\geq x$
- \* Find count of  $> x$
- \* With a fixed-size array, with no looping
- In descending array
  - \* Find some  $= x$
  - \* Find first  $\leq x$
  - \* Find last  $\geq x$
  - \* Find first  $< x$
  - \* Find last  $> x$
  - \* Find count of  $= x$
  - \* Find count of  $> x$
  - \* Find count of  $\geq x$
  - \* Find count of  $\leq x$
  - \* Find count of  $< x$
- Binary search with recursion.
- Binary search with loop.
- Binary search for the root of a function.
  - With loop.
  - With recursion.
- Binary search for a local minimum of a function.
  - With a loop.
  - With recursion.
- Sets, multisets, bags.
- Sorting, definition with sequences and multisets.
- Selection sort.
- Insertion sort.
- Quicksort.



- Lomuto quicksort.
- Dijkstra quicksort.
- Hoare quicksort.
- Yaroslavskiy quicksort (dual-pivot).
- Kushagra quicksort (triple-pivot).
- And perhaps other variants.
- Mergesort.
- Data invariants.
- Binary search trees.
  - Search tree conditions.
  - Binary search trees as sets.
  - Binary search trees as multisets.
  - Data invariant versus pre= and postconditions.
  - Search, with loop and with recursion.
    - \*  $\text{Some} = x$
    - \*  $\text{First} = x$
    - \*  $\text{Last} = x$
    - \*  $\text{First} \geq x$
    - \*  $\text{Last} \leq x$
    - \*  $\text{First} > x$
    - \*  $\text{Last} < x$
  - Insertion.
  - Deletion.
  - AVL trees.
    - \* AVL condition.
    - \* Data invariant versus pre= and postcondition.
  - AA trees.
  - Red=black trees.
  - 2-3 trees.
- Singly linked lists.
  - Reversing a list.
  - Concatenating lists.

- Filtering lists.
- Insertion into a sorted list.
- Singly linked lists in Dafny.
- Mergesort for singly linked lists.
- Quicksort for singly linked lists.
- Samplesort.
- Multiway mergesort.
- Parallel samplesort.
  - With multiple superscalar binary search.
- Parallel multiway mergesort.

## 5 Vikuleg verkefni

Í flestum vikum verða eitt einstaklingsverkefni og eitt hópverkefni. Fyrir hópverkefnin má fjöldi nemenda vera hver sem er, þ.m.t. einn þótt ekki sé mælt með því.

## 6 Weekly Assignments

In most weeks there will be one individual assignment and one group assignment. For the group assignments any number of students can form a group, including a single student, although that is not recommended.

## 7 Lágmrör verkefnaskila

Fyrir lok áttundu viku þurfa nemendur að hafa skilað átta verkefnum samtals. Að öðrum kosti er nemandinn skráður úr námskeiðinu, nema hann eða hún sannfæri kennara um að nemandinn hafi erindi til að halda áfram.

## 8 Minimum Assignment Counts

Before the end of the eighth week students must have turned in eight assignment solutions. Otherwise the student will be deregistered from the course unless he or she convinces the teacher that the student has a chance of passing the course.

## 9 Einkunnagjöf

Bestu sjö einstaklingsverkefni og bestu sjö hópverkefni gilda til einkunnar og vega 50% á móti endanlegri prófseinkunn, sem vegur þá 50%. Einstaklingsverkefnin vega 25% og hópverkefnin vega 25%. Miðmisserispróf verður haldið nálægt áttundu viku. Mæting í það verður valfrjálst en fyrir þá sem mæta í það getur einkunnin úr því hækkað prófseinkunn. Ef nemandi tekur miðmisserisprófið og tekur lokaprófið og einkunnin úr miðmisserisprófinu er hærri en á lokaprófinu þá er endanleg prófseinkunn reiknuð sem vengið meðaltal af miðmisseriseinkunn og lokaprófseinkunn þar sem miðmisseriseinkunnin vegur 40%. Ef endanleg prófseinkunn er fallseinkunn þá mun sú einkunn verða lokaeinkunnin og verkefnaeinkunnir verða þá ekki reiknaðar inn.

## 10 Grade Computations

The best seven individual assignment solutions and the best seven group assignment solutions will be part of the final grade and weigh 50% against the final exam grade, which in turn weighs 50%. The individual assignments weigh 25% and the group assignments weigh 25%. There will be a midterm exam close to the eighth week. Attendance will be optional but for those choosing to take it the grade from the midterm exam may increase the final exam grade. If a student takes the midterm and the grade in the midterm is higher than the grade in the final exam then the final examination grade is recomputed as a weighted mean of the midterm and the final where the midterm weighs 40%. If the final examination grade is a failing grade then that grade will become the final grade for the course and the assignment grades will not be weighed in.

## 11 Fyrirlestrar og æfingatímar

Við munum hittast vikulega á mánudagsmorgnum og nota þann tíma að hluta fyrir fyrirlestra og að hluta fyrir æfingar. Við munum einnig hafa fyrirlestra í Panopto á Canvas, til að hafa meiri tíma fyrir æfingar og verkefni á mánudögum. Allir fyrirlestrar verða teknir upp í Panopto, en ekki allir æfingatímar.

## 12 Lectures and Exercises

We will meet weekly on Monday mornings and use that time partly for lectures and partly for exercises. We will also have some lectures in Panopto on Canvas, in order to have more time for exercises on Mondays. All lectures will be recorded in Panopto, but not all exercise sessions.

## 13 Almennt ítarefni

Vísað verður á ítarefni í vikublöðum. Tilvísanir vikunnar eru eftirfarandi.

- Í Canvas má finna bækling `rokjava.pdf` um rökstudda forritun í Java.
- Finna má ýmislegt varðandi forritunarmálið Dafny sem við byrjum strax að nota. Mælt er með að nota forritunarumhverfið Visual Studio Code og setja upp Dafny í því umhverfi. Einnig má nota Dafny af skipanalínu.
  - Visual Studio Code<sup>3</sup>
  - Dafny niðurhal, o.fl<sup>4</sup>
- Á vefsíðunni <https://tio.run/#dafny> er hægt að skrifa Dafny forritstexta og þýða.
- Youtube rásin Verification Corner<sup>5</sup> með Rustan Leino og félögum.
  - 0:** Basics of specification and verification: Lecture 0<sup>6</sup>, Pre- and postconditions
  - 1:** Basics of specification and verification: Lecture 1<sup>7</sup>, loop invariants
  - 2:** Basics of specification and verification: Lecture 2<sup>8</sup>, binary search
  - 3:** Basics of specification and verification: Lecture 3<sup>9</sup>, the Dutch National Flag algorithm — tengist Dijkstra quicksort sem við munum ræða seinna — góð umfjöllun fyrir utan skýringarmyndina fyrir fastayrðingu lykkju sem er gölluð og villandi
- **Dæmi á Youtube** (ekki Dafny heldur Spec#, sem er svipað): Meira um fastayrðingar lykkju<sup>10</sup> (loop invariants) ásamt reikningum á runu af  $n^3$  gildum án margföldunar – Rustan Leino

## 14 Reference Material

The weeklies will contain references to external resources. This weeks references are the following.

- In Canvas we can find a booklet `rokjava.pdf` (in Icelandic) on reasoned programming in Java.
- Various material regarding the programming language Dafny can be found on the web. We will immediately start using Dafny. It is recommended to use the development environment Visual Studio Code and install Dafny into that environment. Dafny can also be used from the command line.

---

<sup>3</sup><https://code.visualstudio.com/Download>

<sup>4</sup><https://dafny.org/>

<sup>5</sup><https://www.youtube.com/channel/UCP2eLEql4tROYmIYm5mA27A>

<sup>6</sup>[https://www.youtube.com/watch?v=oLS\\_y842fMc](https://www.youtube.com/watch?v=oLS_y842fMc)

<sup>7</sup>[https://www.youtube.com/watch?v=J0FGb6PyO\\_k](https://www.youtube.com/watch?v=J0FGb6PyO_k)

<sup>8</sup>[https://www.youtube.com/watch?v=-\\_tx3lk7yn4](https://www.youtube.com/watch?v=-_tx3lk7yn4)

<sup>9</sup><https://www.youtube.com/watch?v=dQC5m-GZYbk>

<sup>10</sup><https://www.youtube.com/watch?v=spcfzbisBv4>

- Visual Studio Code<sup>11</sup>
- Dafny download and more<sup>12</sup>
- On the web page <https://tio.run/#dafny> it is possible to write and compile Dafny code.
- The Youtube channel Verification Corner<sup>13</sup> with Rustan Leino and colleagues.
  - 0:** Basics of specification and verification: Lecture 0<sup>14</sup>, Pre- and postconditions
  - 1:** Basics of specification and verification: Lecture 1<sup>15</sup>, loop invariants
  - 2:** Basics of specification and verification: Lecture 2<sup>16</sup>, binary search
  - 3:** Basics of specification and verification: Lecture 3<sup>17</sup>, the Dutch National Flag algorithm — tengist Dijkstra quicksort sem við munum ræða seinna — góð umfjöllun fyrir utan skýringarmyndina fyrir fastayrðingu lykkju sem er gölluð og villandi
- An example on Youtube** (not Dafny but rather Spec#, which is similar): More on loop invariants<sup>18</sup> along with computations of a sequence of  $n^3$  values without multiplication – Rustan Leino

## 15 Efni vikunnar

Við munum fjalla um grundvallaratriði rökfræði og röksemdafærslu forrita og einnig sjá dæmi í Dafny. Finnið glærur vikunnar í Canvas og horfið á myndskaið vikunnar í Panopto/Canvas.

## 16 This Weeks Topics

We will discuss the basics of logic and program validation and also see examples in Dafny. Find the slides of the week in Canvas and watch the videos of the week in Panopto/Canvas.

## 17 Verkefni

Skilið lausnum verkefnanna sem PDF í Gradescope innan þeirra tímamarka sem skilgreind eru þar. Sendið einnig lausnina í tölvupósti til mín ([snorri@hi.is](mailto:snorri@hi.is)) eða setjið stutt permalink á lausnina fremst í lausnina.

<sup>11</sup><https://code.visualstudio.com/Download>

<sup>12</sup><https://dafny.org/>

<sup>13</sup><https://www.youtube.com/channel/UCP2eLEql4tROYmIYm5mA27A>

<sup>14</sup>[https://www.youtube.com/watch?v=oLS\\_y842fMc](https://www.youtube.com/watch?v=oLS_y842fMc)

<sup>15</sup>[https://www.youtube.com/watch?v=J0FGb6PyO\\_k](https://www.youtube.com/watch?v=J0FGb6PyO_k)

<sup>16</sup>[https://www.youtube.com/watch?v=-\\_tx3lk7yn4](https://www.youtube.com/watch?v=-_tx3lk7yn4)

<sup>17</sup><https://www.youtube.com/watch?v=dQC5m-GZYbk>

<sup>18</sup><https://www.youtube.com/watch?v=spcfzbisBv4>

**Einstaklingsverkefni.** Notið Dafny til að sanna formúluna

$$\sum_{k=1}^n (2k - 1) = n^2$$

p.e.

$$1 + 3 + 5 + \dots + (2n - 1) = n^2$$

með því að klára forritstextann í skránni `SumOdds-skeleton.dfy` sem þið finnið í Canvas. Þið munið þá hafa sannað formúluna á þrjá vegu, sem allir eru eitthvert afbrigði af vélrænni þrepasönnun. Til hliðsjónar er ráðlegt að lesa glærur viku 1 (einnig í Canvas) þar sem sannað er að

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

p.e.

$$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

**Hópverkefni.** Klárið forritunina í skránni `LinearSearch-skeleton.dfy`, sem þið finnið í Canvas, á tvo vegu, með lykkju og með endurkvæmni. Ekki má breyta lýsingu fallsins sem skilgreint er að öðru leyti en því að leyfilegt er að bæta við „decreases“ klausu.

Í báðum þessum verkefnum þurfið þið að sjá til þess að Dafny þýðandinn samþykki forritstextann, sem tryggir þá að öll föllin virka samkvæmt sinni lýsingu.

## 18 Assignments

Turn in the solutions as PDF files in Gradescope within the timeframe defined there. Also send the solution to me via email ([snorri@hi.is](mailto:snorri@hi.is)) or put a short permalink to the solution at the beginning of the solution.

**Individual Assignment.** Use Dafny to prove the formula

$$\sum_{k=1}^n (2k - 1) = n^2$$

i.e.

$$1 + 3 + 5 + \dots + (2n - 1) = n^2$$

by finishing coding the program in the file `SumOdds-skeleton.dfy` which you will find in Canvas. You will then have proved the formula in three ways, which are all some variant of computerized induction. For reference it is useful to read the slides of the week (also in Canvas) where it is proven that

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

i.e.

$$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

**Group Assignment.** Finish the programming in the file `LinearSearch-skeleton.dfy`, which you will find in Canvas, in two ways, with a loop and with recursion. The description of the function may not be changed except that it is allowed to add a "decreases" clause.

In both these assignments you must ensure that the Dafny compiler accepts the source code, which then ensures that the functions work according to their descriptions.