1 Course intro

Important:

- Register Teams.
- Make a Team (if this is not possibly add name to Excel)

Why do we care? We want to deliver a high quality program. Maybe the program contains mistakes. Some mistakes can have severe consequences.

Foundations of software verification

- How to specify what constitutes "correct behaviour"
- How to verify the correctness of the program
- What constitutes good tests? When have we tested enough?
- Can we automate this?

Learning goals: Concepts that represent different aproaches: Pragmatism (testing) vs completeness. **Not in scope:** Project management **Pre-requisite:** C-sharp, Ide, set-theory, logic **Project & assignment:** 3 homeworks, 3 assignments, 2 testing projects with 3 people **Grading:** half is exam **Software:** Jetbrains Raider IDE, Github (private) **Load:** 16hrs/week

2 Unit testing

Verifying the program by inspectin a finite number of executions. This is pragmatic and incomplete.

Unit should not be too large

We will be using NUnit.

3 Project

Implementing a game (Nethack).

4 Graphs

Paths need to be consecutive. Test paths need to start at the entry point. Length of a path is the amount of edges.