



Stichting NIOC

Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2025, gehouden op donderdag 27 maart 2025 jl. en georganiseerd door Hogeschool Windesheim). Bij elkaar zo'n 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op 18 maart 2027 in Arnhem en wordt georganiseerd door HAN University of Applied Sciences.

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

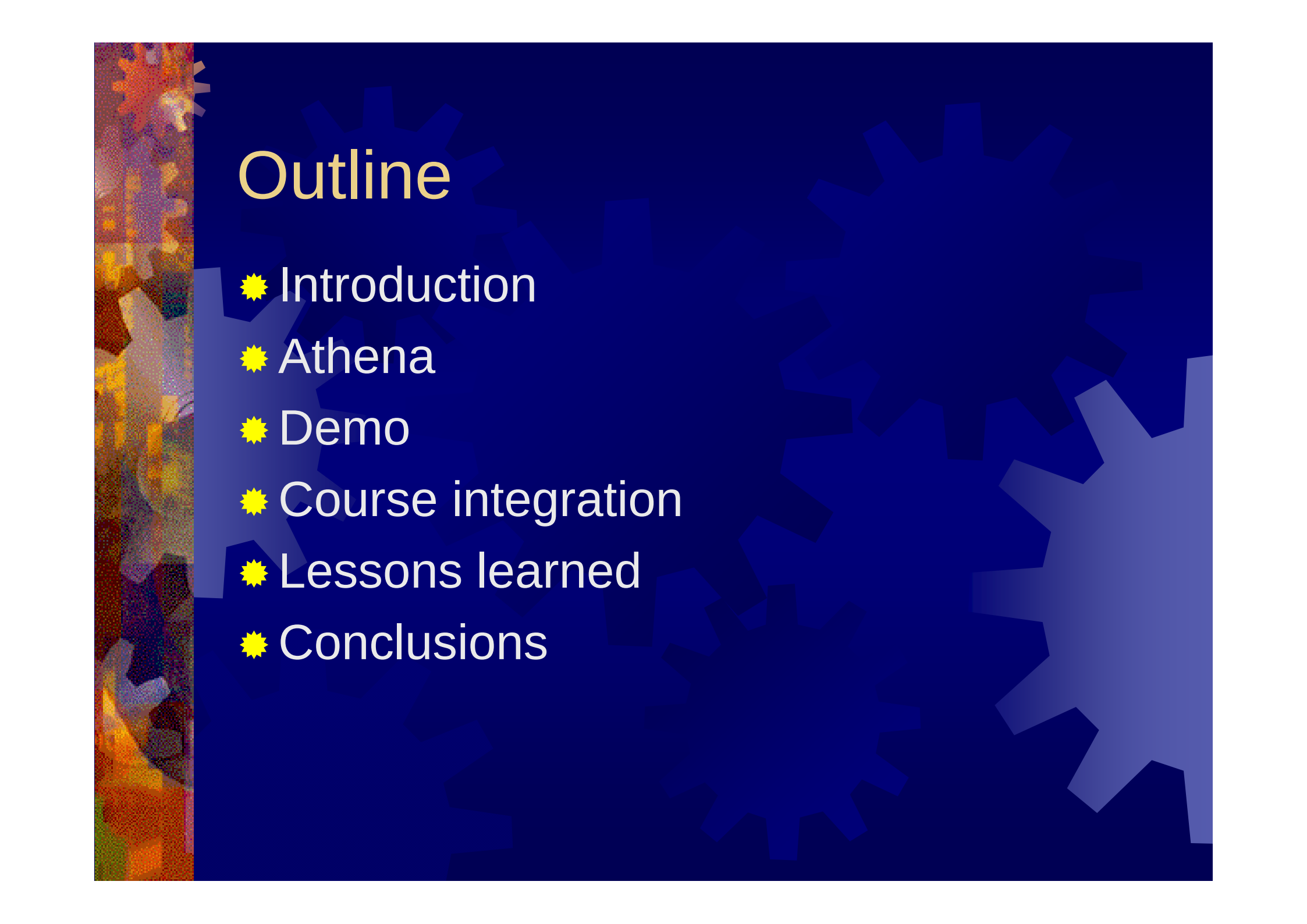
Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.



Athena, a large scale programming lab support tool

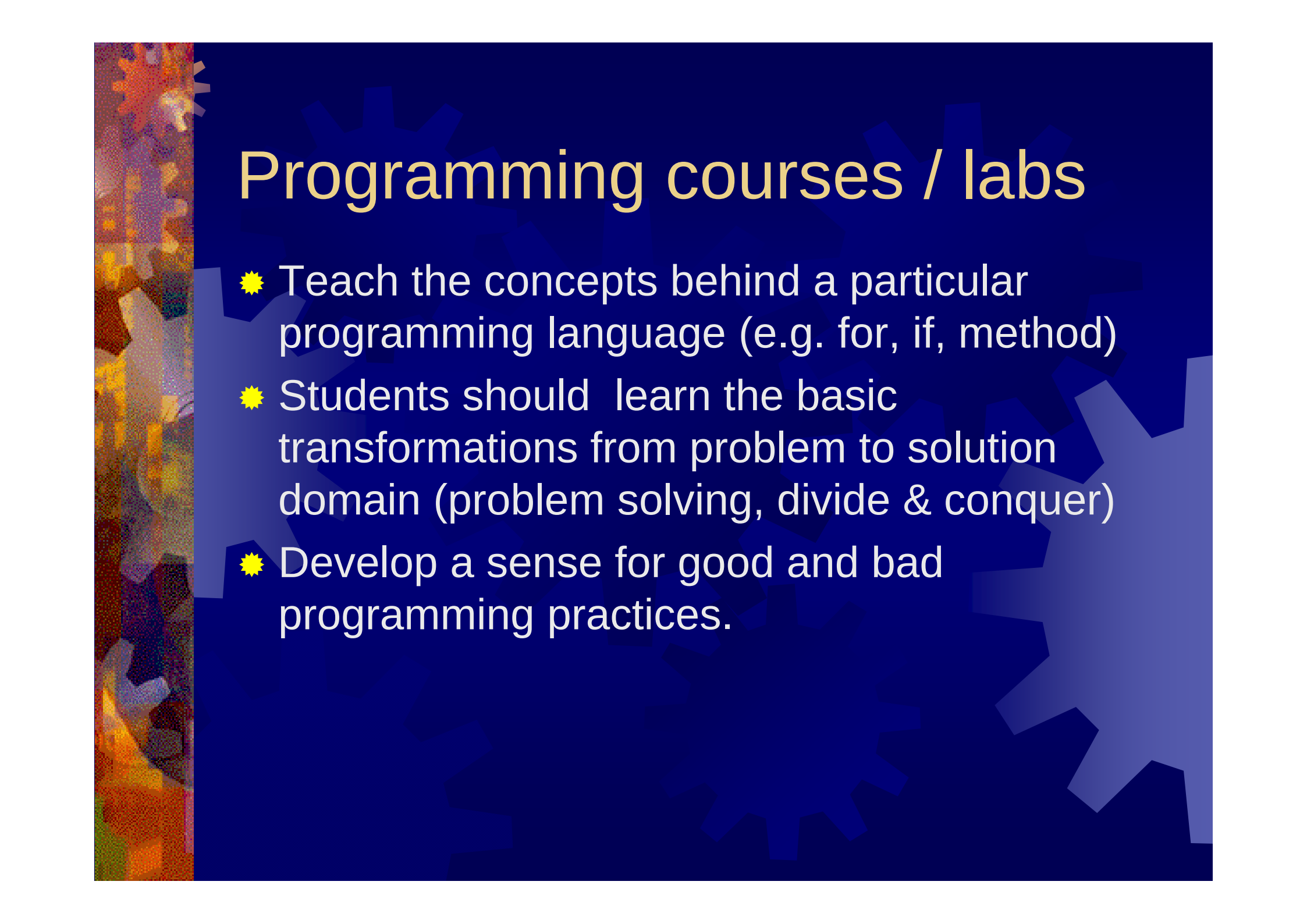
Anton Jansen, Ph.D. Student

Software Engineering and ARCHitecture
(SEARCH) University of Groningen
The Netherlands



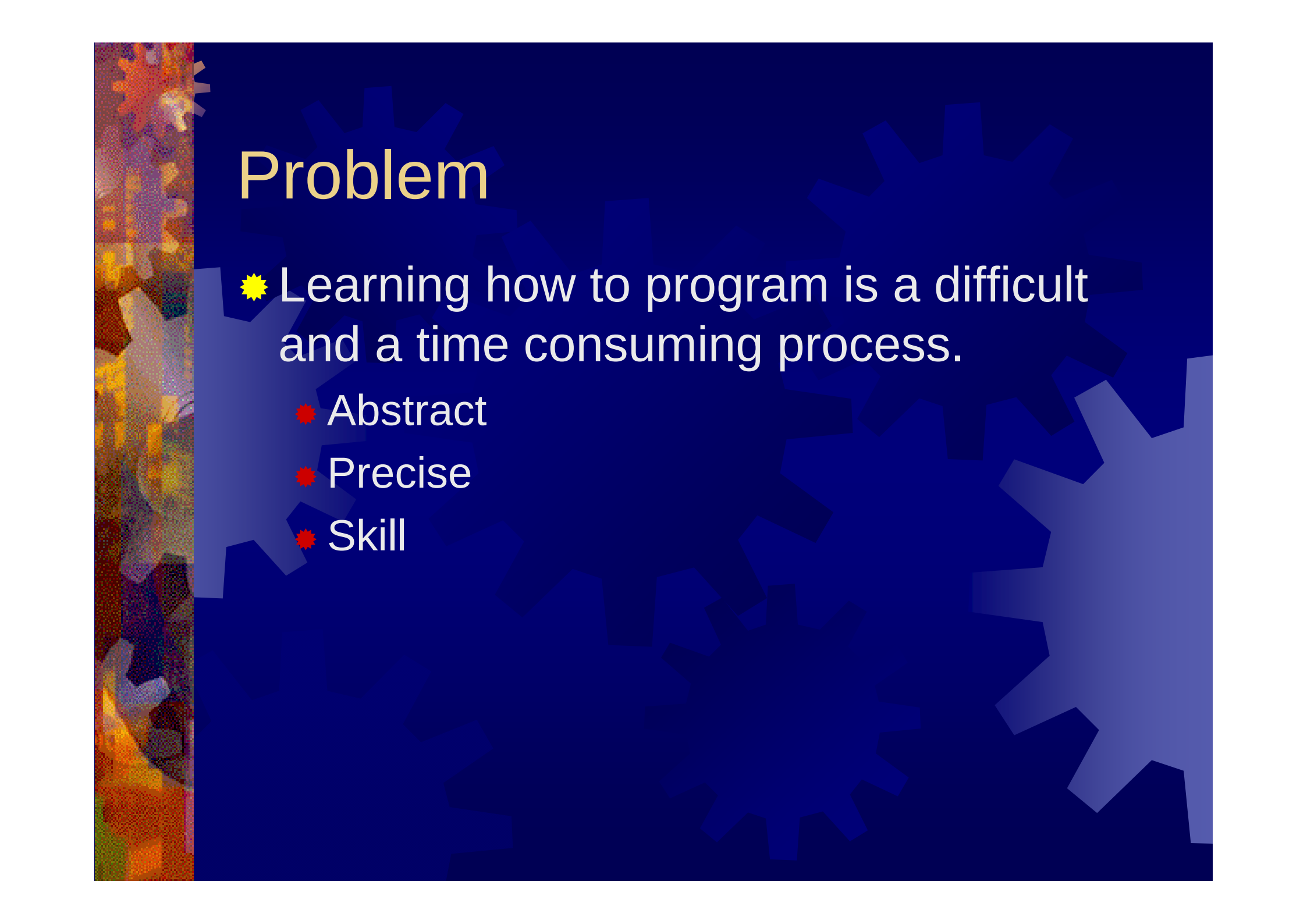
Outline

- ✦ Introduction
- ✦ Athena
- ✦ Demo
- ✦ Course integration
- ✦ Lessons learned
- ✦ Conclusions



Programming courses / labs

- ✴ Teach the concepts behind a particular programming language (e.g. for, if, method)
- ✴ Students should learn the basic transformations from problem to solution domain (problem solving, divide & conquer)
- ✴ Develop a sense for good and bad programming practices.



Problem

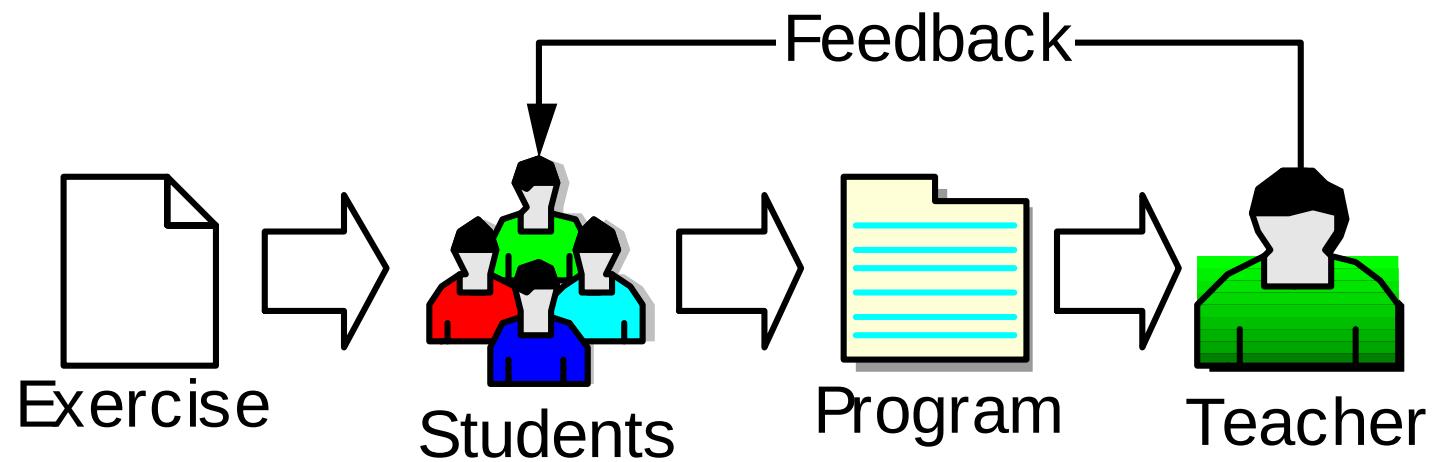
✦ Learning how to program is a difficult and a time consuming process.

- ✦ Abstract
- ✦ Precise
- ✦ Skill

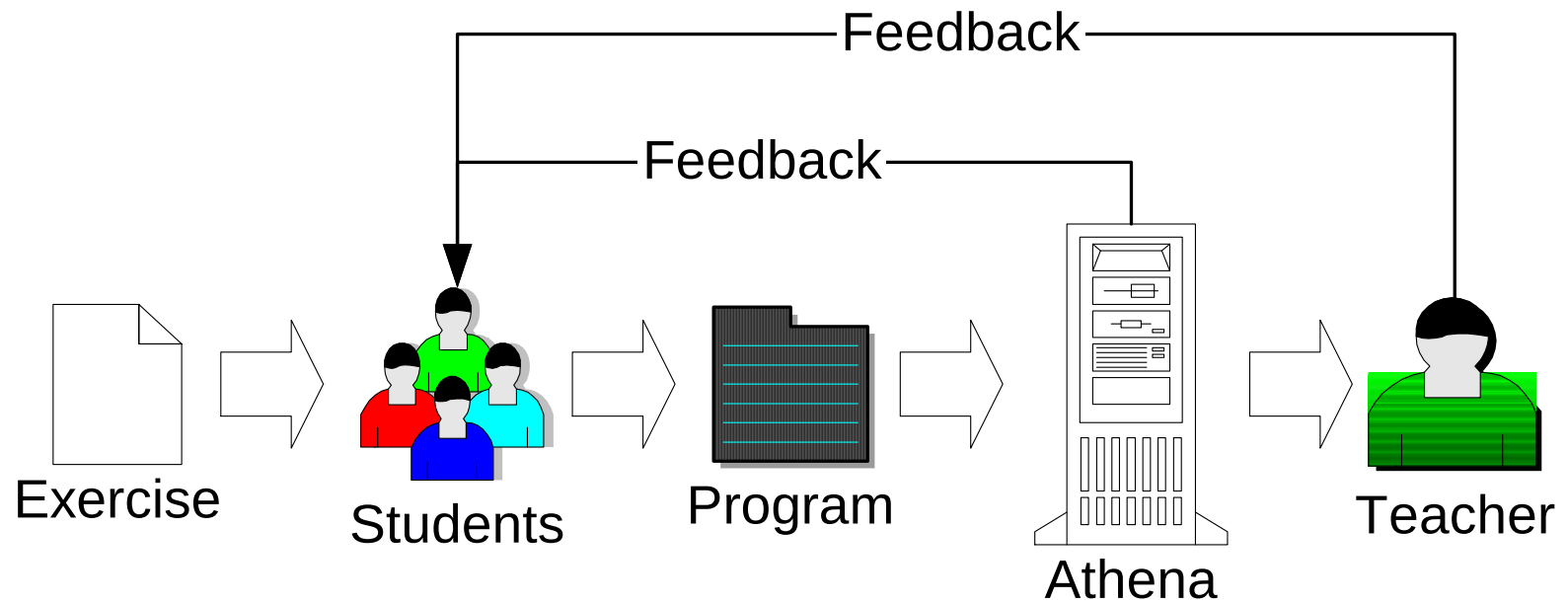
Labs

- ✱ Programming exercises to train special use of concepts and combinations of them
- ✱ Need many small exercises to train concepts in isolation and particular combinations
- ✱ => A lot of work for the staff to judge all these solutions (e.g. 7.5kLOC for 60 students)

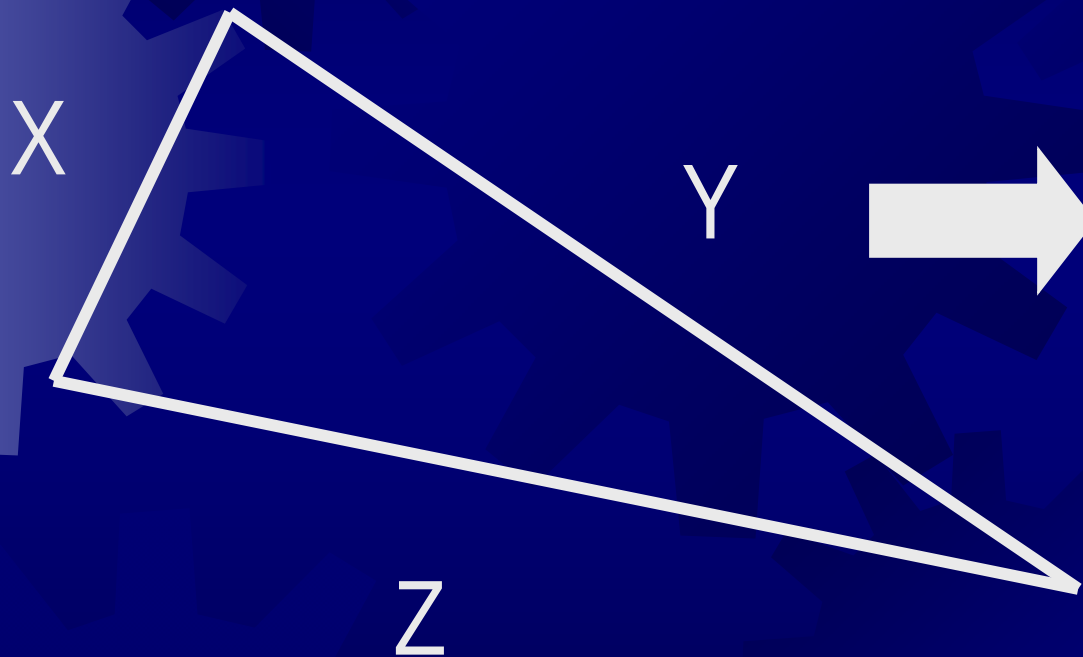
Programming lab exercises



Athena



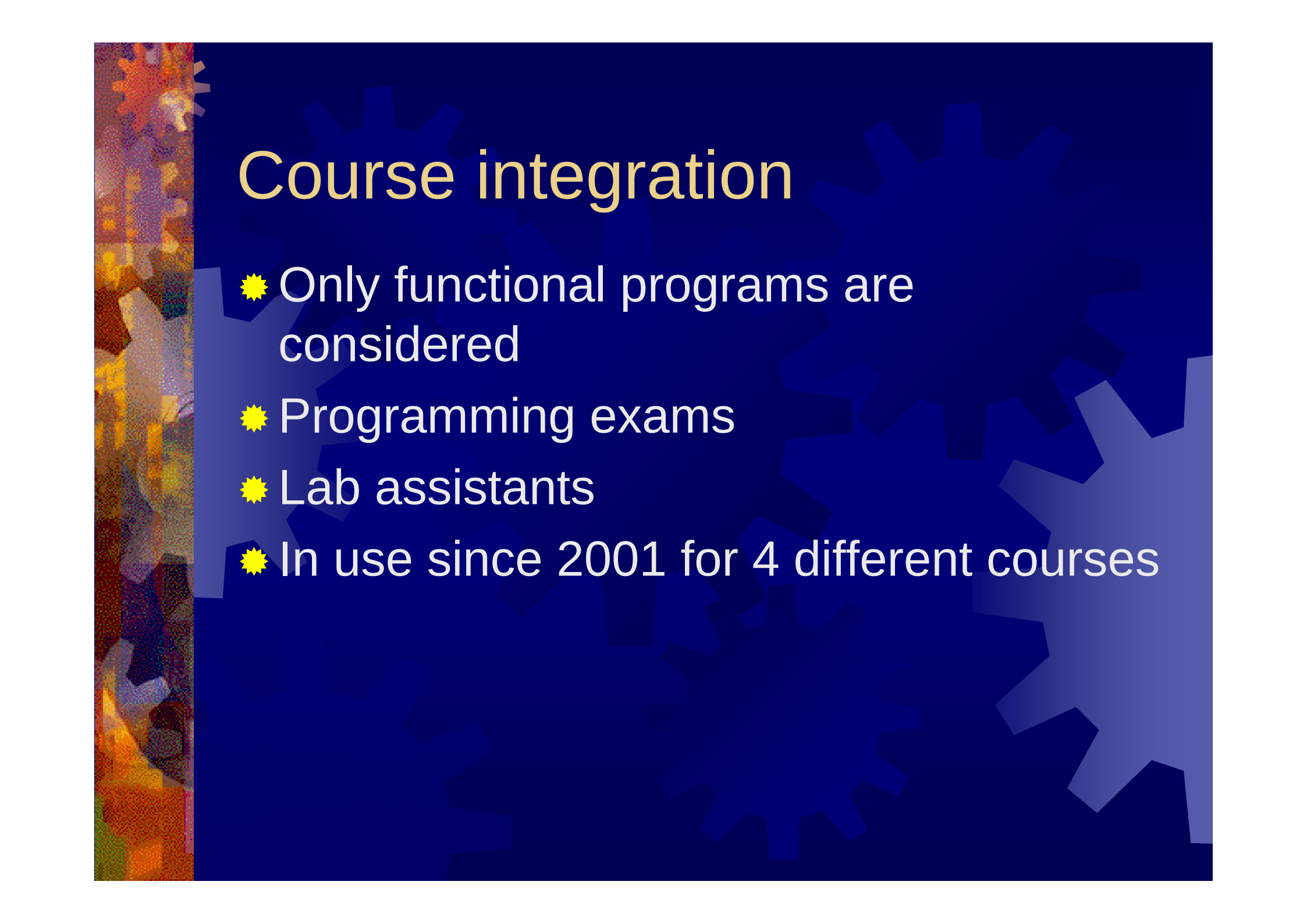
Example exercise: a triangle



- Sharp
- Right-angled
- Obtuse-angled
- Impossible

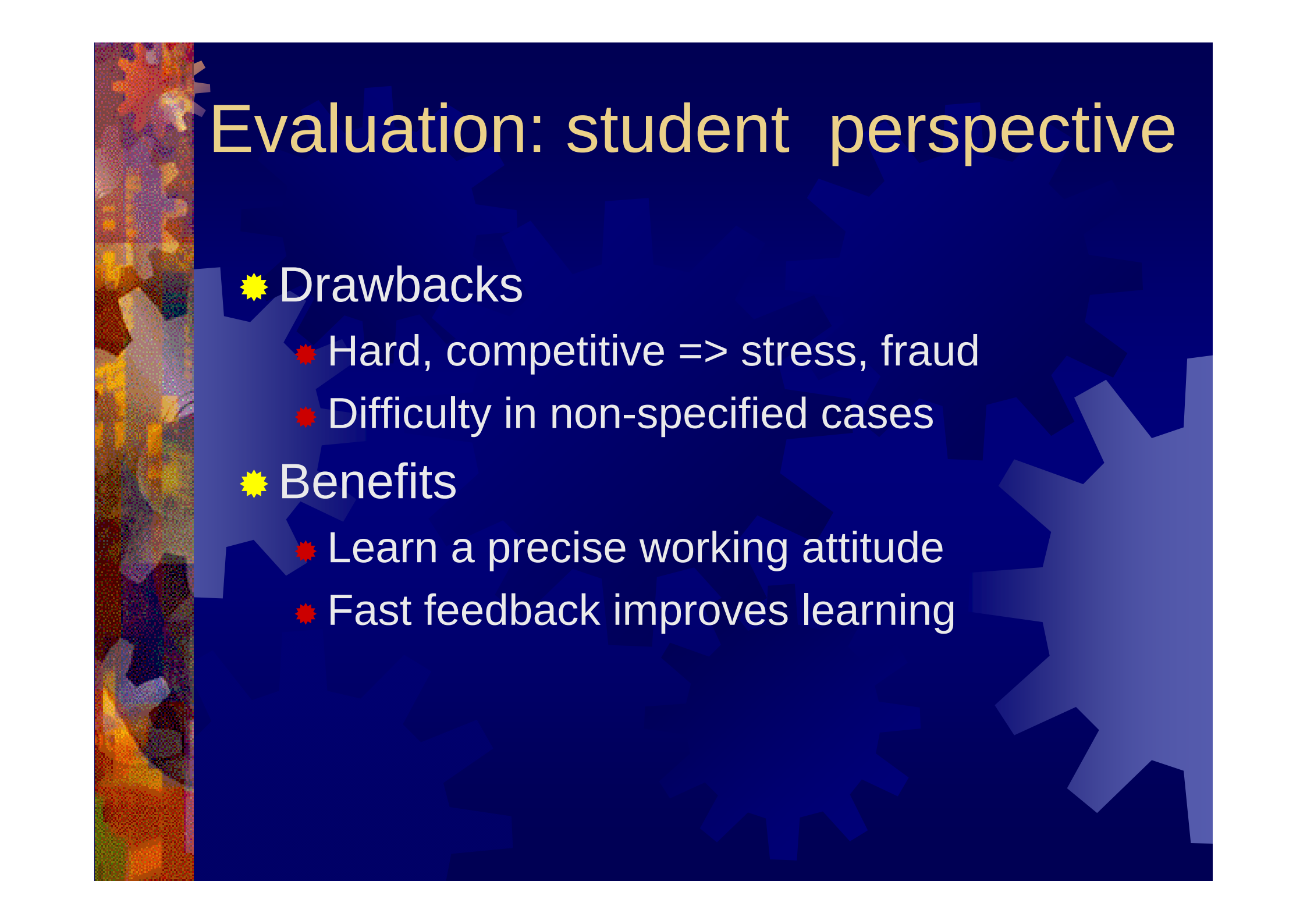
Automated testing

- ✱ Don't do it
- ✱ Compile
- ✱ Run
- ✱ Output testing
 - ✱ Textual differences
 - ✱ Numerical differences
 - ✱ Dedicated test program
- ✱ Performance testing (time, memory, etc.)
- ✱ Let students make the test themselves



Course integration

- ✴ Only functional programs are considered
- ✴ Programming exams
- ✴ Lab assistants
- ✴ In use since 2001 for 4 different courses



Evaluation: student perspective

☀ Drawbacks

- Hard, competitive => stress, fraud
- Difficulty in non-specified cases

☀ Benefits

- Learn a precise working attitude
- Fast feedback improves learning



Evaluation: teacher perspective

☀ Drawbacks

- ☀ Dependability

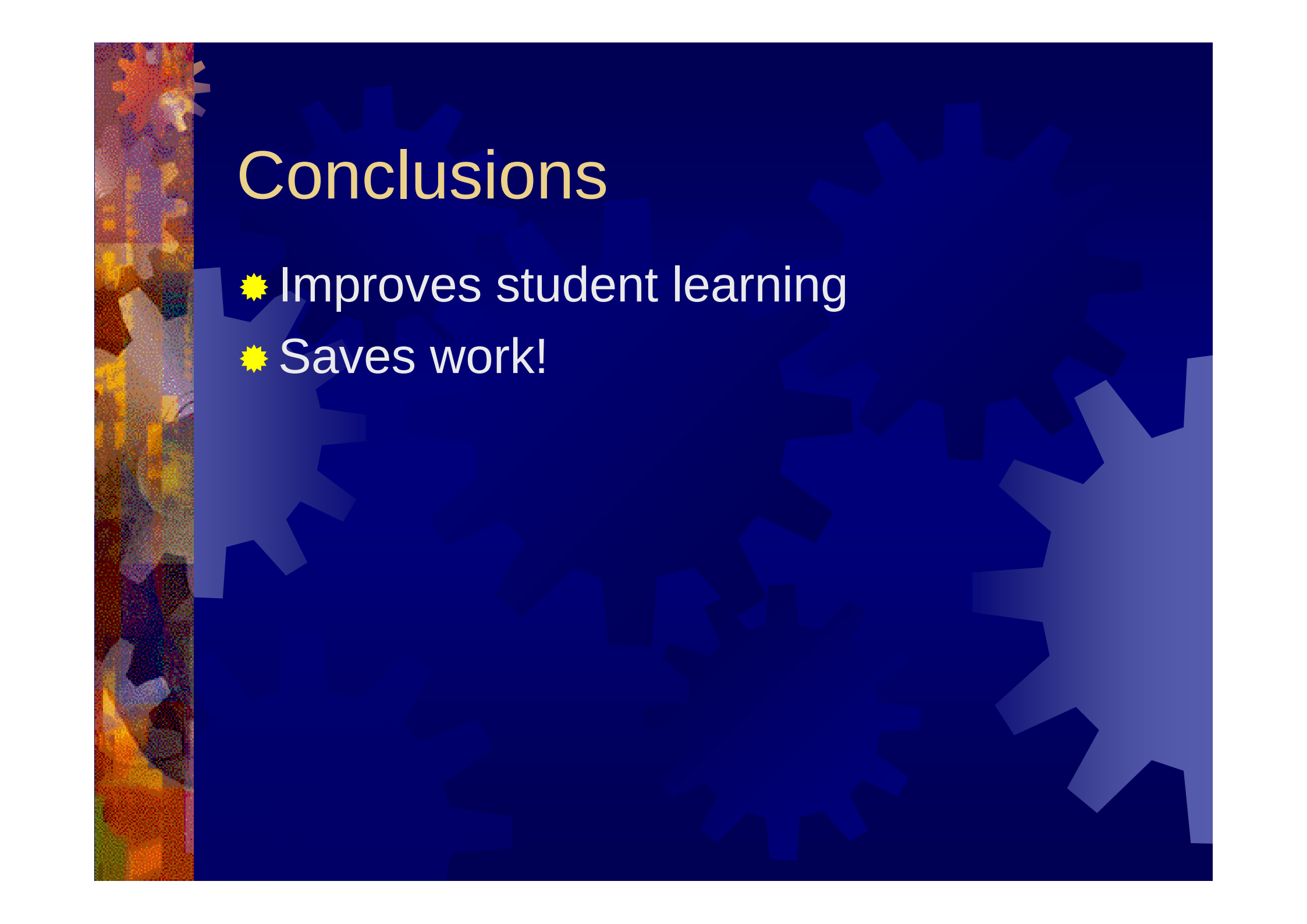
☀ Benefits

- ☀ Shifting focus from functionality to quality
- ☀ Initial investment high, but pays off



Athena key features

- ✦ Language & platform independent
- ✦ Point and click interface
- ✦ Multiple courses
- ✦ Any place, any time



Conclusions

- ✱ Improves student learning
- ✱ Saves work!



Questions?