#12 CODERUNNER IN THE TEACHCENTER

Experiences with and application of the programming question type

Challenges for beginner programmers

As with other fields, the best way to learn programming is through practice. However, especially as a beginner, it is often challenging to set up the right development environment.

Questions such as "What is a compiler?", "What is a terminal?" or "How do I run my programme?" need to be answered before you can even start programming.

This often leads to beginners feeling overwhelmed, as at this point the actual learning process of programming has not even begun.

What is CodeRunner?

This is exactly where the TeachCenter plugin CodeRunner comes in. As a question type for quizzes, it allows students to write, execute and test code directly in the TeachCenter without having to worry about setting up a local development environment.

> *The instructions on TELucation explain how teachers can create such questions.*

[1]

Once students have written their code, it is automatically checked against predefined test cases. Students receive immediate feedback and can see how well their solution works. This automated feedback not only saves valuable time for teachers, but also promotes independent learning and problem-solving skills among students.

CodeRunner in practice

> *Short description of the course "Object-Oriented Programming 1"*
>
> **Type:** design exercise (KU)
>
> **Group size:** ~400
>
> **Teachers:** Lukas Eberhard, Sebastian Gürtl, David Kerschbaumer
>
> **Institute:** Institute of Human-Centred Computing (HCC)

CodeRunner was used as part of the "Object-Oriented Programming 1" course to familiarise students with a new programming language.

This course is offered in the second semester of all computer science programmes and is the second major programming lecture for many students. After gaining initial programming experience in the first semester, students now switch to the new programming language C++. For this reason, it is important to keep the entry requirements as low as possible and give students the opportunity to try out the new concepts immediately.

As part of the first homework assignment in week 2, an activity with four CodeRunner tasks was created.

Students were able to submit their solutions in the TeachCenter and received points for them.

In order to create a learning environment without an exam situation, students were able to work on the examples at any time and revise their solutions as often as they wanted during the two-week period.

Fig. 1: Task 1 of the homework assignment in OOP1. Students see the description of the task and an example of how the code is tested. Below that, they are asked to find and test the solution themselves.

Practical advice

CodeRunner offers a wide range of settings in the TeachCenter, which can seem overwhelming for teachers at first. A few settings have helped to make the most of CodeRunner.

- Customising test cases: Teachers can influence how test cases are executed. This also helps to better understand the general logic.

- Setting grade and deductions: The option to deduct points for each incorrect attempt has been deactivated. It could tempt students to programme on external platforms and only use CodeRunner for the final submission.

- Modifying input field for students: Detailed documentation can be found online. The following settings have proven effective for this course:

```
{
    "auto_switch_light_dark" : true,
    "font_size" : "15px",
    "live_autocompletion" : true,
    "theme" : "terminal"
}
```

[2]

- Entering sample solution: By writing a sample solution in the answer field, the stored test cases are automatically compared with the sample solution. This helps identify errors in the test cases or in the solution quickly.
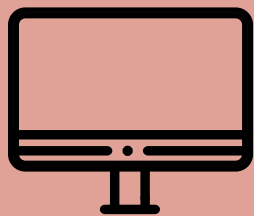
1 https://telucation.tugraz.at/en/programming-questions-coderunner/

2 https://trampgeek.github.io/moodle-qtype_coderunner/#ui-parameters

#12

CODERUNNER IN THE TEACHCENTER

Experiences with and application of the programming question type

Author	David Kerschbaumer (Institute of Human-Centred Computing)
Date of Publication	10/2025
License	

Use cases in teaching

CodeRunner offers many opportunities for integration into teaching. Its greatest potential lies in teaching the basics of programming and providing immediate practice through small tasks. This can be combined perfectly with interactive learning materials: topics are first covered in theory in the TeachCenter through texts or videos and then reinforced through a CodeRunner task.

CodeRunner also offers opportunities for interdisciplinary application. CodeRunner can offer considerable added value, especially in courses that frequently involve calculations but where a separate programming course is not feasible. Teachers provide the programming environment in the TeachCenter so that students can focus on the essentials: the code for calculating the solution.

Another possible scenario is to use CodeRunner in exams and quizzes. Teachers can automatically grade submissions using the test cases. At the same time, students can receive feedback on whether their solution is correct already during the exam, if desired. This not only saves teachers a lot of time, but also allows students to assess their performance during the exam.

Limitations and our strategy with CodeRunner

The integration of CodeRunner has received very positive feedback from students, as it is ideal for smaller projects. However, the limitations of the tool quickly become apparent in larger projects. It does not offer the option of debugging code, and the output of values via print functions is unfortunately not very clear. Nevertheless, the tool comes at just the right time.

Last year, "weekly exercises" were introduced – small, weekly compulsory tasks that summarise the most important points from the lecture and must be implemented. Until now, an external system was used for this, which led to numerous problems and additional work. Starting in the winter semester of 2025, the weekly exercises will be integrated directly into the TeachCenter with CodeRunner.

However, it should be noted that tasks at this level (code with less than 50 lines) can now be solved in seconds using large language models such as ChatGPT.

Further information on possible applications of AI in teaching can also be found in the Didactics section in the article: #10 Use of AI-based tools in teaching (Part 2)

It is therefore important to make it clear to students that the task is not to generate the solution as quickly as possible, but to find their own solutions and improve their own skills.