

# #12 CODERUNNER IM TEACHCENTER

## Erfahrungsbericht und Einsatzmöglichkeiten des Programmier-Fragetyps

### Herausforderungen für Programmieranfänger:innen

Ähnlich wie bei anderen Fachgebieten lernt man das Programmieren am besten durch Üben. Gerade als Anfänger:in ist es jedoch oft eine Herausforderung, die richtige Entwicklungsumgebung einzurichten.

Fragen wie „Was ist ein Compiler?“, „Was ist ein Terminal?“ oder „Wie führe ich mein Programm aus?“ müssen geklärt werden, bevor man überhaupt mit dem Programmieren beginnen kann.

Dies führt bei Anfänger:innen häufig zu Überforderung, da zu diesem Zeitpunkt der eigentliche Lernprozess des Programmierens noch gar nicht begonnen hat.

### Was ist CodeRunner?

Genau hier setzt das TeachCenter-Plugin CodeRunner an. Als Fragetyp für Tests ermöglicht es den Studierenden, direkt im TeachCenter Code zu schreiben, auszuführen und zu testen, ohne sich um die lokale Einrichtung einer Entwicklungsumgebung kümmern zu müssen.

**i** Wie solche Fragen von Lehrenden erstellt werden können, wird in der Anleitung auf TELucation erklärt.



Sobald die Studierenden ihren Code geschrieben haben, wird dieser automatisch gegen vordefinierte Testfälle überprüft. Die Studierenden erhalten dabei sofortiges Feedback und sehen, wie gut ihre Lösung funktioniert. Dieses automatisierte Feedback spart nicht nur wertvolle Zeit für die Lehrenden, sondern fördert auch das eigenständige Lernen und die Problemlösungskompetenz der Studierenden.

### CodeRunner im Praxistest

**Kurzsteckbrief der Lehrveranstaltung**  
**„Objektorientierte Programmierung 1“**

**Lehrveranstaltungstyp:** Konstruktionsübung

**Anzahl Studierende:** ~400

**LV-Leitung:** Lukas Eberhard, Sebastian Gürtl, David Kerschbaumer

**Institut:** Institute of Human-Centred Computing (HCC)



Im Rahmen der Lehrveranstaltung „Objektorientierte Programmierung 1“ (OOP1) wurde CodeRunner eingesetzt, um Studierende mit einer neuen Programmiersprache vertraut zu machen.

1 <https://telucation.tugraz.at/programmierfragen-coderunner/>

Dieser Kurs wird im zweiten Semester aller Informatikstudiengänge angeboten und ist für viele Studierende die zweite große Programmiervorlesung. Nachdem sie im ersten Semester erste Programmiererfahrungen gesammelt haben, wechseln die Studierenden nun zur neuen Programmiersprache C++. Aus diesem Grund ist es wichtig, die Einstiegshürde so gering wie möglich zu halten und Studierenden die Möglichkeit zu geben, die neuen Konzepte unmittelbar auszuprobieren.

Im Zuge der ersten Hausübung in Woche 2 wurde eine Aktivität mit vier CodeRunner-Aufgaben erstellt.

Die Studierenden konnten im TeachCenter ihre Lösung einreichen und erhielten dafür auch Punkte.

Um eine Lernumgebung ohne Prüfungssituation zu schaffen, konnten die Studierenden während der zweiwöchigen Laufzeit jederzeit und beliebig oft an den Beispielen arbeiten und ihre Lösungen überarbeiten.

### Tipps und Tricks aus der Praxis

CodeRunner bietet im TeachCenter eine Vielzahl von Einstellungsmöglichkeiten, was für Lehrende anfangs schwer überschaubar wirken kann. Ein paar Einstellungen haben geholfen, CodeRunner optimal zu nutzen:

- Ausführung der Testfälle anpassen: Lehrende können beeinflussen, wie die Testfälle ausgeführt werden. Dies hilft auch dabei, die generelle Logik besser zu verstehen.
- Bewertung und Abzüge einstellen: Die Option, für jeden falschen Versuch einen Anteil der Punkte abzuziehen, wurde deaktiviert. Sie könnte Studierende verleiten, auf externen Plattformen zu programmieren und CodeRunner nur für die finale Abgabe zu nutzen.

- Eingabefeld für Studierende modifizieren: Eine detaillierte Dokumentation ist online zu finden. Für diesen Kurs haben sich folgende Einstellungen bewährt:

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

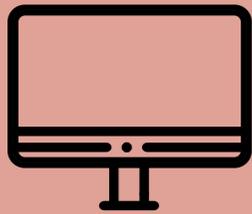
- Musterlösung hinterlegen: Durch das Hinterlegen einer Musterlösung im Antwortfeld werden die hinterlegten Testfälle automatisch mit der Musterlösung getestet. So können Fehler in den Testfällen oder in der Lösung schnell erkannt werden.

2 [https://trampgeek.github.io/moodle-gtype\\_coderunner/#ui-parameters](https://trampgeek.github.io/moodle-gtype_coderunner/#ui-parameters)



Abb. 1: Aufgabe 1 der Hausübung in OOP1. Studierende sehen die Beschreibung der Aufgabe und ein Beispiel, wie der Code getestet wird. Darunter soll die Lösung selbst erarbeitet und getestet werden.





# #12

## CODERUNNER IM TEACHCENTER

Erfahrungsbericht und Einsatzmöglichkeiten des Programmier-Fragetyps

<b>Autor</b>	David Kerschbaumer (Institute of Human-Centred Computing)
<b>Veröffentlichung</b>	10/2025
<b>Lizenz</b>	 Dieser Beitrag wird unter den Vorgaben der Lizenz Creative Commons Attribution 4.0 International (siehe <a href="https://creativecommons.org/licenses/by/4.0/deed.de">https://creativecommons.org/licenses/by/4.0/deed.de</a> ) zur Verfügung gestellt.

### Einsatzmöglichkeiten in der Lehre

CodeRunner bietet viele Möglichkeiten, in die Lehre integriert zu werden. Das größte Potenzial liegt in der Vermittlung von Programmiergrundlagen und dem unmittelbaren Üben anhand kleiner Aufgaben. Dies lässt sich hervorragend mit interaktiven Lernmaterialien kombinieren: Themen werden im TeachCenter zuerst theoretisch durch Texte oder Videos behandelt und anschließend durch eine CodeRunner-Aufgabe gefestigt.

Darüber hinaus bietet CodeRunner auch große interdisziplinäre Einsatzmöglichkeiten. Besonders in Lehrveranstaltungen, in denen häufig Berechnungen anfallen, ein eigener Programmierkurs aber zu aufwendig wäre, kann CodeRunner einen erheblichen Mehrwert bieten. Lehrende stellen die Programmierumgebung im TeachCenter bereit, sodass sich die Studierenden auf das Wesentliche konzentrieren können: den Code zur Berechnung der Lösung.

Ein weiteres spannendes Anwendungsgebiet ist der Einsatz von CodeRunner in Klausuren und Tests. Lehrende können die Abgaben automatisch mithilfe der Tests bewerten. Gleichzeitig erhalten die Studierenden wenn gewünscht schon während der Prüfung Feedback, ob ihre Lösung korrekt ist. Dies spart nicht nur den Lehrenden enorm viel Zeit, sondern ermöglicht es den Studierenden auch, ihre Leistung bereits während der Prüfung einzuschätzen.

### Grenzen und unsere Strategie mit CodeRunner

Die Integration von CodeRunner hat bei den Studierenden sehr positives Feedback erhalten, da es sich ideal für kleinere Projekte eignet. Bei größeren Projekten sind die Limitierungen des Tools jedoch schnell erreicht. Es bietet keine Möglichkeit, Code zu debuggen, und die Ausgabe von Werten über print-Funktionen ist leider nicht sehr übersichtlich. Trotzdem kommt das Tool genau zum richtigen Zeitpunkt.



Im vergangenen Jahr wurden sogenannte „Weekly Exercises“ eingeführt – kleine, wöchentliche Pflichtaufgaben, die das Wichtigste aus der Vorlesung zusammenfassen und implementiert werden müssen. Bisher wurde dafür ein externes System genutzt, was zu zahlreichen Problemen und Mehraufwand führte. Ab dem Wintersemester 2025 werden die Weekly Exercises direkt im TeachCenter mit CodeRunner integriert.



Weitere Informationen über Einsatzmöglichkeiten von KI in der Lehre finden Sie auch in der Rubrik **Didaktik** im Beitrag: #10 Einsatzmöglichkeiten von KI-gestützten Tools in der Lehre (Teil 2)

Dabei muss jedoch beachtet werden, dass Beispiele dieser Größenordnung (Implementierungen mit weniger als 50 Zeilen) heutzutage mithilfe von Large Language Models wie ChatGPT innerhalb von Sekunden gelöst werden können.



Es ist daher wichtig, den Studierenden klarzumachen, dass die Aufgabe nicht darin besteht, die Lösung schnellstmöglich generieren zu lassen, sondern sie selbst zu erarbeiten und damit die eigenen Fähigkeiten zu verbessern.