SCIENTIX
The community for science education in Europe
CISCO
STE(A)M PARTNERSHIPS
Education Resilience in Europe
©Marina Zlochin - stock.adobe.com
vector - stock.adobe.com
European Schoolnet
SCIENTIX

The STE(A)M Partnerships has been funded under the European Union's H2020 research and innovation programme – project Scientix 4, coordinated by European Schoolnet (EUN). The content of the document is the sole responsibility of the organizer and it does not represent the opinion of the European Commission (EC), and the EC is not responsible for any use that might be made of information contained.

STEAM EmbRaCe

A STEAM project for Empathy, Resilience and Creativity

Introduction to Programming

Author(s)	
Kechagias Andreas, Giannaros Ilias	
Summary	
This course is designed to provide students with a comprehensive introduction to programming using Python. It will cover the basics of programming concepts, including variables, data types, conditional statements, loops, and functions. Throughout this course, students will collaborate in groups, developing and implementing their acquired skills in projects with real - world applications. This course is suitable for students of all experience levels, offering a strong foundation for future careers in disciplines such as development, engineering, and technology.	
Key elements	
Keywords	Programming fundamentals / Python / Data types / Control structures / Functions / Modules
Subject	Computer Science / Mathematics
Age of students	12 - 17
Preparation time	8 hours
Teaching time	4 - 6 hours
Online teaching material	-
Offline teaching material	Steam EmbRaCe “Intro to Programming” presentation
Resources used	-

Licenses

Trends

Artificial Intelligence / Machine Learning / Data Science / Automation / Web Development / Internet of Things (IoT)

21st century skills

Creativity / Critical Thinking / Problem Solving / Collaboration and Teamwork / Technology Literacy

Lesson Plan

Activity	Description	Duration
Introduction to algorithms and programming	Introduce yourself and your background in programming to initiate a conversation about the students' comprehension of the topic, its significance and get to know their experience in programming so as to direct the lesson accordingly. Discuss the key concepts of algorithmic thinking and programming. Ask fundamental questions on the topic such as what is a program and an algorithm, what is the difference between these concepts, why programming is important etc. Once most of the students have gotten involved and expressed their points of view, use the Steam EmbRaCe "Intro to Programming" presentation to analyze and clarify the concepts discussed.	45 min
Software demonstration	Provide a brief introduction to the platform that will be used so that the class is familiar with its fundamental features. Preferably, use a free online platform that is simple for students to utilize (indicatively https://replit.com).	15 min
Introduction to programming fundamentals	Use the Steam EmbRaCe "Intro to Programming" presentation to explain the basic elements of programming using Python. Discuss data types, variables, conditional statements, loops, and functions. In order to illustrate the programming concepts being taught, it's important to provide examples and encourage active participation from the class through the platform that was presented. Basic, exemplary problems should be presented that are adaptable based on the level of the class. If the class is made up of students with differing levels of experience, it may be necessary to design different tasks to ensure that all students are appropriately challenged. Allow time for questions and clarification. It can be helpful to brainstorm real - world scenarios with the class to demonstrate how the concepts can be applied in practice in a real - world setting.	90 min
Implementation	To reinforce the programming concepts taught in the lesson, it is recommended to prepare projects that require the implementation of as many of these concepts as possible. Enable small group work among the students to improve collaboration and communication. Give them enough time	90 min

Lesson Plan

Activity	Description	Duration
	to examine the problem, encourage them to design an algorithmic solution and offer guidance while they develop their code. (It is advised that the difficulty of the problems be scalable.) Groups should be given the chance to present their solutions and get feedback on both the outcome and the procedure they followed to get there after the assignment has been completed.	
Recap and review	Summarize the key concepts and skills learned in the course. Allow time for further discussion. Emphasize the importance of practice and continued learning in programming.	15 min

SEL practices

The following techniques support self-awareness and self-management which are the two main domains of the CASEL model in social and emotional learning.

At the beginning of the course we identify students' emotional state by following the activity "Practice for identifying emotional state".

At the end of the lesson students reflect upon their work by following the activity of Reflection.

After the reflection they practice the square breathing technique and the aim is for them to learn to practice this every time they are about to begin a challenging activity.

Assessment

Use the following exercises of graded difficulty for student assessment:

1. Create an interactive trivia using Kahoot! on the terms and concepts discussed in class.
2. Create an interactive hot spot exercise using H5P where the student is required to identify syntax errors in given code snippets.
3. Create an interactive missing word exercise using Wordwall where the student is required to type in the missing keywords (reserved words), special symbols, operands and/or numbers in order for given code snippets to function properly.

About STEAM EmbRaCe project

This Learning Scenario has been created in the framework of the STEAM EmbRaCe project.

STE(A)M EmbRaCe aims to promote inclusion by engaging and inspiring students from different backgrounds. Students work on real-world STE(A)M problems, which will help develop their cultural empathy, resilience, and creative thinking. The idea is to create digital content which will be ready to be used by teachers in any classroom setting. More specifically, the project will allow the development of a 7-week course and teacher training on how to use the developed material with students.

Find out more about the STEAM EmbRaCe project:

https://steamingthefuture.gr/steam-embrace/

Annex 1

Official Python documentation:
https://docs.python.org/3.9/index.html

Official Python Beginner’s Guides:
https://wiki.python.org/moin/BeginnersGuide

Official Python community forum:
https://python-forum.io/

Use the following Python tutorials to familiarize yourself and your students with the Python programming language:
https://www.w3schools.com/python/
https://www.tutorialspoint.com/python/index.htm