(http://csmatters.org) 2 - 1

0b10 - 0b1

Programming: Introduction and Motivation

Unit 2. Developing Programming Tools

Revision Date: Jan 16, 2020
Duration: 2 50-minute sessions

Lesson Summary

Summary

Students will be introduced to programming for the first time. They will learn about computer science, computing for good, some of the potential outcomes of programming, and the definition of abstraction. Students will also be learning about program design.

Learning Objectives

CSP Objectives

- *EU CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.*
    - LO CRD-2.B - Explain how a program or code segment functions.
    - LO CRD-2.C - Identify input(s) to a program.
    - LO CRD-2.E - Develop a program using a development process.
    - LO CRD-2.F - Design a program and its user interface.
    - LO CRD-2.G - Describe the purpose of a code segment or program by writing documentation.
    - LO CRD-2.I - For errors in an algorithm or program: a. Identify the error. b. Correct the error.
- *EU DAT-1 - The way a computer represents data internally is different from the way the data is interpreted and displayed for the user. Programs are used to translate data into a representation more easily understood by people.*
    - LO DAT-1.A - Explain how data can be represented using bits.
- *EU AAP-1 - To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.*
    - LO AAP-1.A - Represent a value with a variable.

    - LO AAP-1.B - Determine the value of a variable as a result of an assignment.

Common Core ELA:

- RST 12.4 - Determine the meaning of symbols, key terms, and other domain-specific words and phrases
- RST 12.10 - Read and comprehend science/technical texts
- WHST 12.2 - Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes
- WHST 12.9 - Draw evidence from informational texts to support analysis, reflection, and research

Key Concepts

Students must understand that stereotypes of computer programmers are not accurate and that 'coding' is something everyone can learn to do. Computing can be a creative expression and used for good.

Possible misunderstandings: The term NGO is used in the article *Programming for Good: The Story of Code for India*, without explicitly defining it as Non-Governmental Organization.

Abstraction is a tricky idea. Python allows us to describe what we want to do such as "print" and "input" because it provides the details that explain to the computer how to accomplish the task of taking many keypresses followed by a press of the Enter key to allow the computer to store the information we entered, and also knows how to take information stored in the computer's memory and cause it to appear as a series of recognizable dots on the screen using print. Computer programs use input, output, processing, and memory.

The design topics taught in the 2020-2021 session introduce both user interaction and user experience design.

Outcomes

- Students will describe various creative and helpful purposes for programming.
- Students will define and give examples of abstraction.
- Students will create Python code in Runestone.
- Students will be able to describe how diverse user input and collaboration improve the design process.

Essential Questions

- How can computing and the use of computational tools foster creative expression?
- How does abstraction help us in writing programs, creating computational artifacts and solving problems?
- How can computation be employed to help people process data and information to gain insight and knowledge?
- What opportunities do large data sets provide for solving problems and creating knowledge?
- How are programs developed to help people, organizations or society solve problems?
- How are programs used for creative expression, to satisfy personal curiosity or to create new knowledge?
- How does abstraction make the development of computer programs possible?
- How does computing enhance human communication, interaction, and cognition?
- How does computing enable innovation?
- How do economic, social, and cultural contexts influence innovation and the use of computing?

Teacher Resources

Student computer usage for this lesson is: **required**

TEACHERS need to have the class and user accounts set up in Runestone to track student progress.

- **http://interactivepython.org/runestone/static/thinkcspy/toc.html (http://interactivepython.org/runestone/static/thinkcspy/toc.html) Create your own classroom**

For the Students:

- *Programming allows us to solve problems and use computing for good* articles used in this lesson:
  - http://www.attendly.com/programming-for-good-the-story-of-code-for-india (http://www.attendly.com/programming-for-good-the-story-of-code-for-india/)
- *Seven things you should know when you're starting out programming* article:
  - http://www.theguardian.com/info/developer-blog/2011/oct/07/programming-developer-journalist (http://www.theguardian.com/info/developer-blog/2011/oct/07/programming-developer-journalist)
- "Meet a computer scientist" to see the diveristy in computer science
  - http://mcwic.github.io/htmlblocks/computerscientistlibrary.html (http://mcwic.github.io/htmlblocks/computerscientistlibrary.html)
- Creating a Student Account in Runestone (Make sure to add your Runestone class number before printing)
  - https://docs.google.com/document/d/1FlyM5-jHIcDWCTOCnInS3ZDNinAECRybqoCqTc03VzQ/edit?usp=sharing (https://docs.google.com/document/d/1FlyM5-jHIcDWCTOCnInS3ZDNinAECRybqoCqTc03VzQ/edit?usp=sharing)

Note: If computer is not being used, students will need their own copies of the articles

Lesson Plan

Session 1

Getting Started (5 min)

[use the Presentation on Programming Python in Runestone]

Journal Entry: What are some ways that writing programs is a creative endeavor? [slide 2]

Share answers with your elbow partner. Then share answers with the class.

Guided Activities (40 min)

Activity 1: Computing for Good [15 min]

Open Discussion: What are some ways you know that computing has been used for "good?" [slide 3]

Go to *Programming for Good: The Story of Code for India*

http://www.attendly.com/programming-for-good-the-story-of-code-for-india/ (http://www.attendly.com/programming-for-good-the-story-of-code-for-india/)

Read articles or selected text in pairs, with alternating pairs each reading one article. Pairs of pairs get together to share what they read and what they got out of the article.

In pairs, answer the following questions:

1. How did the impact of Hurricane Sandy in 2012 begin the concept of Code for India?

2. What is the reason for the Adopt-a-School app, and what is the result of its use?
3. What is the reason for the Spotter app?
4. What is the reason for the Bravehearts app, and what is its significance for public safety? Where else would this app be useful?
5. If you were able to design an app "for good," what problem would you try to solve?

Check for Understanding: Teams should share their answers to their instructor.

Activity 2: Coding with Python in Runestone [25 min]

Have students sketch the computer hardware architecture diagram in Python for Everybody Chapter
See https://books.trinket.io/pfe/01-intro.html#computer-hardware-architecture (https://books.trinket.io/pfe/01-intro.html#computer-hardware-architecture).

Explain that computer processes guide the flow of information that is input, processed, stored or output. The processor (CPU) uses short term memory for doing calculations and temporary storage and long term memory if needed. Have students sketch the computer components diagram above. Explain to students that they will learn to program to develop software using each of these components.

The program output is usually based on both input values and values already in memory. Watch the following video to see how this occurs.

Watch the video on how programs use input, processing, memory, and output to run programs.
https://www.youtube.com/watch?v=BSTrWhAGta8 (https://www.youtube.com/watch?v=BSTrWhAGta8)

Read Computer Hardware architecture by Dr. Severance.

https://books.trinket.io/pfe/01-intro.html#computer-hardware-architecture (https://books.trinket.io/pfe/01-intro.html#computer-hardware-architecture)

.

BE SURE YOU HAVE SET UP STUDENT ACCOUNTS

1. [Use the Presentation on Programming Python in Runestone slides 4]. Open Runestone as was done in Unit 1 Lesson 6. Students use their accounts to access the eTextbook
2. Read The Way of the Program and answer questions on the student handout.
3. Go to the next section: Algorithms and check to see that all student answers are being correctly recorded in Runestone. Display answer results on the teacher screen. Have students continue to fill in answers on the student handout. [slide 5]
4. Have students independently complete the next 3 sections (or finish them for homework) and fill in the student handout.
5. Skip ahead to A Typical First program [slides 7-13]
    1. Demonstrate how to run the program, change the code, add an input line, and display the value input in the print statement.
    2. Explain the idea of syntax, and the result of various mistakes.
6. Discuss abstracting in coding. [slide 14]

Homework: Seven things you should know if you're starting out programming

Have students go to the article *Seven things you should know if you're starting out programming* at

http://www.theguardian.com/info/developer-blog/2011/oct/07/programming-developer-journalist (http://www.theguardian.com/info/developer-blog/2011/oct/07/programming-developer-journalist)

Pre-reading activities:

1. Before dividing the students into groups for the next activity, it is important to address the paragraph at the beginning of the reading, which mentions the "coder stereotype" and includes a parenthetical note that the author believes this stereotype to be (largely) accurate. Explain that because of the breadth of application areas of computing in today's world, this stereotype is not an accurate view of the diverse field of computing – there are a wide variety of people in computing and a wide variety of applications of computing education. See videos at http://mcwic.github.io/htmlblocks/computerscientistlibrary.html#top (http://mcwic.github.io/htmlblocks/computerscientistlibrary.html#top)
2. Point out that a program development process can be ordered and intentional with a specific goal in mind, or exploratory in nature depending on the situation.

In pairs or groups of three, assign each group to read and summarize one of the seven programming principles in the article.

1. Logic (not "math")
2. Catch a shooting star (variables)
3. Dictionary (data types)
4. Russian Dolls (things within things, instances)
5. Sausage (processes)
6. The dog, the cat, and the fish (causation, event change)
7. Pizza (abstraction) include why abstraction is "like making pizza," and what other kinds of activities might fall into that category.

Check for Understanding: Each person should write a 140 character tweet on their topic, next class the group should share their findings with the class.

Session 2

Getting Started (5 min)

Journal

Say: Online tools support collaboration by allowing programmers to share and provide feedback on ideas and document. Often this collaboration takes place online. What are some online tools you think programmers might use to collaborate?

Student share responses. Be sure that receiving feedback from both developers and users is discussed.

Guided Activities (40 min) Use the Program Design presentation in the Lesson 1 Resource folder.

Activity 1 - Requirements (10 min)

Say: Success and failure are both about meeting criteria. To know if our efforts are a success, we need to know what the requirements are. So the first step of the design process is investigating the program requirements. In the development process, the design phase outlines how to accomplish a given program specification. Software developers typically organize the overall program into a collection of individual components or modules. Collectively, the component or modules meet the program's requirements.

Highlight: The users are important!

1. Consulting and communicating with users is important to the development of computing innovations.
2. Information gathered from potential users helps to better understand the purpose of a program from diverse perspectives and to develop a program that fully incorporates multiple needs and perspectives.

Highlight: Innovation is enhanced by group collaboration!

1. Developing in a group allows a greater variety of designs and better quality innovation.
    - Quote: ***“If the group is an art form of the future, then convening groups is the artistry we must cultivate to fully explore the promise of this form.”*** *— Fetzer Institute (https://www.creativityatwork.com/2008/05/15/12-ways-to-enhance-creativity-and-collaboration-in-teams/)*
2. Online tools support collaboration by allowing programmers to share and provide feedback on ideas and documents.

Jigsaw: In groups of four, read and discuss the four techniques for identifying requirements as listed by Hans Jonasson **(https://www.pmi.org/learning/library/determining-project-requirements-best-practices-7278)**.

For each of the four techniques, use the Design Techniques table to list the techniques purpose, its advantages, and its limitations

Design Requirements Techniques

Technique	Purpose	Advantage	Disadvantage

Activity 2 - User Interactions (20 min)

Say: User interaction design, at its heart, is all about satisfying the end-user’s needs and wants. The design may include brainstorming, storyboarding or even used interaction design as is this activity.

As a class, watch this short UI/UX Design Process video (https://vimeo.com/93634945) to see how many of the following design strategies are illustrated.

- brainstorming
- planning and storyboarding
- organizing the program into modules and functional components
- creation of diagrams that represent the layouts of the user interface
- development of a testing strategy for the program

Investigate user interface design (UI) https://www.usability.gov/what-and-why/user-interface-design.html (https://www.usability.gov/what-and-why/user-interface-design.html)

Investigate user experience design (UX). User Interaction Foundation (https://www.interaction-design.org/literature/topics/ux-design)

On the board, have students list possible ways that an investigation can be performed and keep going until all of the following methods have been uncovered:

- collecting data through surveys
- user testing
- interviews
- direct observations

Activity 3 Development Phase (10) min

The rest of this unit will address techniques used in the development phase of a program.

Development is both iterative - with multiple rounds of testing and improvement - and incremental with development occurring in small testable steps.

Development then should include documentation throughout its development and support program testing throughout to meet the overall design requirements. Documention is important whether working individually or collaboratively. Documentaion includes comments stored within programs where supported by the programming language (not all languages support comments) and outside the program (such as in web pages).

True/False. For each of the following statements, work with an elbow partner to determine if the statements are true or false.

For the false statements, modify the statements so they are meaningful and true.

> Program documentation describes the function and development of program segments.
>
> Comments written in a program are read by the computer before the program can execute.
>
> Program documentation is best done after the program is written and tested.
>
> For efficiency reasons, program documentation should only be written when working in a collaborative program development environment.
>
> If a programming environment does not support comments then none should be used.
>
> Program documentation comments should be as terse as possible since they are internal notes to the developers.

Bookmark or store your these true statements for future reference.

Wrap Up: (3 min)

Design requires a lot of effort throughout the programming project.

Write a short paragraph explaining to a new project manager the benefits of investing this time in the design process.

Options for Differentiated Instruction

Oral reading strategies such as "popcorn reading" where students take turns reading a paragraph and then pass the reading off to another student in the class, or other reading strategies such as students reading together quietly in pairs, can be used for longer texts.

Longer readings can be broken up by sections or paragraphs to speed the lesson up or keep students engaged.

Activity 1 could be assigned as homework from the day before as a step into this lesson to allow more time in class for the other readings:

Additional activity: Ask students to read the Preface (pages iii-iv) and Chapter 1 introduction and section 1.1 (pages 1-2) of Python for Everybody (http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf) and answer these questions:

1. Why does the author think that Python is a better teaching language for beginning programmers than Java?
2. Why does the author consider programming to be a creative activity?
3. What things are computers better at doing than people?
4. What things are people better at doing than computers?
5. What are some motivations for writing computer programs? List some responses not included in the selected

reading.

6. What do you think of the author's writing style? Name some positive and negative aspects of the presentation in the text.

Evidence of Learning

Formative Assessment

Students will use several different strategies for reading and writing responses based on their readings.

Classroom discussions and student responses (written and oral) will allow instructors to check for understanding.

Summative Assessment

Summative assessment will be included in Part 2 on the use of the PyCharm IDE.

Student handout filled in

Tweet from optional homework assignment

Progress recorded in Runestone for the General Introduction

- The Way of the Program (http://interactivepython.org/runestone/static/thinkcspy/GeneralIntro/intro-TheWayoftheProgram.html)
- Algorithms (http://interactivepython.org/runestone/static/thinkcspy/GeneralIntro/Algorithms.html)
- The Python Programming Language (http://interactivepython.org/runestone/static/thinkcspy/GeneralIntro/ThePythonProgrammingLanguage.html)
- Executing Python in this Book (http://interactivepython.org/runestone/static/thinkcspy/GeneralIntro/SpecialWaystoExecutePythoninthisBook.html)
- More About Programs (http://interactivepython.org/runestone/static/thinkcspy/GeneralIntro/MoreAboutPrograms.html)
- A Typical First Program (http://interactivepython.org/runestone/static/thinkcspy/GeneralIntro/ATypicalFirstProgram.html)

(http://www.umbc.edu/) (http://www.umd.edu/) (http://www.nsf.gov/)

Authored by: CS Matters in Maryland
Website: csmatters.org (http://csmatters.org)
Email: csmattersinmaryland@gmail.com
(mailto:csmattersinmaryland@gmail.com)

by University of Maryland, Baltimore County (http://umbc.edu) and University of Maryland, College Park (http://umd.edu).