

SCHOOL DISTRICT OF THE CHATHAMS

Game Design with Python Grades 9 - 12 Semester

Course Overview

Python provides an introduction to the field of computer science and can be used as a prerequisite for AP Computer Science Principles. The basics of imperative programming will be covered as well as general problem solving principles. We will review the basics of the language, discuss core programming topics, and then spend time looking at some specialized uses, such as video game creation. It also covers general material on the discipline of computer science. Topics include problem solving, the software development process, structured programming, top-down design, and programming language features. These features include identifiers, data types, input/output commands, control flow statements, classes, and user-defined methods.

New Jersey Student Learning Standards

The New Jersey Student Learning Standards (NJSLS) can be located at www.nj.gov/education/cccs/2020/.

Computing Systems

8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.

8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.

Algorithms & Programming

8.1.12.AP.1: Design algorithms to solve computational problems using a combination of original and existing algorithms.

8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.

Technology Standards

9.4.12.DC.6: Select information to post online that positively impacts personal image and future college and career opportunities.

9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities and utility for accomplishing a specified task.

21st Century Integration | NJSLS 9

9.4.12.CI.1: Demonstrate the ability to reflect, analyze and use creative skills and ideas.

9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving.

9.3.IT.2 Use product or service design processes and guidelines to produce a quality information technology product or service.

9.3.IT.12 Demonstrate knowledge of the hardware components associated with information systems.

9.3.IT.13 Compare key functions and applications of software and determine maintenance strategies for computer systems.

- 9.3.IT-PRG.1 Analyze customer software needs and requirements.
9.3.IT-PRG.2 Demonstrate the use of industry standard strategies and project planning to meet customer specifications.
9.3.IT-PRG.3 Analyze system and software requirements to ensure maximum operating efficiency.
9.3.IT-PRG.4 Demonstrate the effective use of software development tools to develop software applications.
9.3.IT-PRG.5 Apply an appropriate software development process to design a software application.
9.3.IT-PRG.6 Program a computer application using the appropriate programming language.
9.3.IT-PRG.7 Demonstrate software testing procedures to ensure quality products.

Career Ready Practices

- CRP2. Apply appropriate academic and technical skills.
CRP4. Communicate clearly and effectively and with reason.
CRP6. Demonstrate creativity and innovation.
CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.
CRP11. Use technology to enhance productivity.

Interdisciplinary Connections

Comprehensive Health & Physical Education

- 2.2.12.LF.4: Exhibit responsible social behavior by including and cooperating with classmates of all skill levels, assisting when needed and collaborating respectfully to solve problems in groups, teams and in pairs.

Units of Study

Unit 1: Introduction to Python using Karel the Robot (~4 days)

- What are the properties of the robot world?
- What are methods?
- What are comments?
- What are the types of errors when writing programs?
- What are the parts of a simple program?
- What are words (identifiers) in Java?

Unit 2: Extending the Robot Programming Language (~8 days)

- What is inheritance?
- How do we define a new method?
- What is a constructor?
- What does the 'super' keyword mean?
- What are the steps in designing a program?
- What is stepwise refinement?

Unit 3: Conditionally Executing Instructions (~8 days)

- How do we write conditional statements using if and if/else?
- What are boolean methods?
- What is the negation operator?
- How do we nest if statements?
- What are transformations of if instructions?
- When can we bottom or top factor?
- When are tests redundant?

Revision Date: November 2021

- How do we write our own boolean methods?

Unit 4: Variables and Strings (~14 days)

- How do we print to output?
- How does concatenation work?
- What are primitive data types?
- What are variables and constants?
- How do we assign a value?
- How do we scan input?
- What are the arithmetic expressions? (introducing mod)
- What is operator precedence?
- What does dynamically typed mean?

Unit 5: Final Projects (~45 days)

- Plan and develop a full python game based on the specifications you outline

<h3>Learning Objectives/Discipline Standards of Practice</h3>

Learning Objectives:

- Create, run and debug simple Python programs using an IDE and Karel the Robot.
- Import and run supplied programs using an IDE.
- Create a world using Karel the Robot
- Understand the anatomy of a typical class – import statements and comments.
- Demonstrate understanding of object creation and class inheritance.
- Demonstrate understanding of the basic syntax and flow of control in Java programs.
- Design programs using quality design methodology.
- Create classes including user defined methods on top of existing classes.
- Understand the anatomy of a typical class – constructors
- Demonstrate understanding of flow of control in Python programs.
- Create programs that manipulate flow of control through if and if/else statements.
- Create programs that manipulate flow of control through while statements
- (Advanced) Create programs that implement user-defined boolean methods
- Create programs that can take, store and manipulate user input.
- Demonstrate understanding of the primitive data types in Python.
- Create programs that can manipulate data types where necessary.
- Design and implement a Python game referencing the techniques learned through the semester

Discipline Standards of Practice:

- Computing Systems
 - People interact with a wide variety of computing devices that collect, store, analyze, and act upon information in ways that can affect human capabilities both positively and negatively. The physical components (hardware) and instructions (software) that make up a computing system communicate and process information in digital form.
- Networks and the Internet
 - Computing devices typically do not operate in isolation. Networks connect computing devices to share information and resources and are an increasingly integral part of computing. Networks and communication systems provide greater connectivity in the computing world.

- Impacts of Computing
 - Computing affects many aspects of the world in both positive and negative ways at local, national, and global levels. Individuals and communities influence computing through their behaviors and cultural and social interactions, and, in turn, computing influences new cultural practices.
- Data & Analysis
 - Computing systems exist to process data. The amount of digital data generated in the world is rapidly expanding, so the need to process data effectively is increasingly important. Data is collected and stored so that it can be analyzed to better understand the world and make more accurate predictions.
- Algorithms & Programming
 - An algorithm is a sequence of steps designed to accomplish a specific task. Algorithms are translated into programs, or code, to provide instructions for computing devices. Algorithms and programming control all computing systems, empowering people to communicate with the world in new ways and solve compelling problems.
- Engineering Design
 - People design for enjoyment and to solve problems, extend human capabilities, satisfy needs and wants, and improve the human condition. Engineering Design, a systematic approach to creating solutions to technological problems and finding ways to meet people's needs and desires, allows for the effective and efficient development of products and systems.
- Interaction of Technology and Humans
 - Societies influence technological development. Societies are characterized by common elements such as shared values, differentiated roles, and cultural norms, as well as by entities such as community institutions, organizations, and businesses. Interaction of Technology and Humans concerns the ways society drives the improvement and creation of new technologies, and how technologies both serve and change society.
- Nature of Technology
 - Human population, patterns and movement focus on the size, composition, distribution, and movement of human populations and how they are fundamental and active features on Earth's surface. This includes understanding that the expansion and redistribution of the human population affects patterns of settlement, environmental changes, and resource use. Patterns and movements of population also relate to physical phenomena including climate variability, landforms, and locations of various natural hazards and their effects on population size, composition, and distribution.
- Effects of Technology on the Natural World
 - Many of engineering and technology's impacts on society and the environment are widely regarded as desirable. However, other impacts are regarded as less desirable. Effects of Technology on the Natural World concerns the positive and negative ways that technologies affect the natural world.
- Ethics & Culture
 - Ethics and Culture concerns the profound effects that technologies have on people, how those effects can widen or narrow disparities, and the responsibility that people have for the societal consequences of their technological decisions.

Instructional Resources and Materials

Whole class resources have been identified with an asterisk.

Resources

- *Monty Karel: A Gentle Introduction to the Art of Object-Oriented Programming in Python* by Joseph Bergin
- Pygames Documentation & Tutorials
- IDE Documentation
- Python Documentation

Materials

- Desktop Computing Devices

Assessment Strategies

Assessment is designed to measure a student's mastery of a course standard and learning objective. Assessment can be used for both instructional purposes (formative assessment) and for evaluative purposes (summative assessment).

The following is a general list of the many forms assessment may take in learning.

- Peer Feedback
- Self-Reflection
- Reflective Exit Tickets/Slips

Course Specific Assessments Include:

- (Karel) Programming
- Pygames