

PUBLIC SCHOOLS OF EDISON TOWNSHIP
OFFICE OF CURRICULUM AND INSTRUCTION



Functional Programming in Python

Length of Course:	Term
Elective/Required:	Elective
Schools:	High School
Eligibility:	Grade 9-12
Credit Value:	6 Credits
Date Approved:	August 23, 2022

TABLE OF CONTENTS

Statement of Purpose	3
Suggested Time Schedule	5
Unit 1: Introduction to Computers	6
Unit 2: Basics of Programming	7
Unit 3: Conditional Statements	8
Unit 4: Loops	9
Unit 5: User Defined Functions	10
Unit 6: Modules	11
Unit 7: Data Structures & Statistics	12

Statement of Purpose

The focus of this course is on learning the functional aspects of programming. This intro-level course will teach students how to utilize the basic structures common to most actionable programming languages in a language with a low-barrier to entry (Python). As a result of this course, students should be able to translate their understanding of the concepts they learned in Python to other programming languages that have the same or similar structures by only learning the new syntax. A concentrated focus of the course will be analytical computing problem solving skills.

This guide was revised by:
Robert Giordano (JPS)
John Krajunas (EHS)

Completed under the supervision of:

A copy of this curriculum guide for review on the District website and in the Office of Curriculum and Instruction.

Course Name:	Functional Programming in Python
Grade Level(s)	9-12
Comp Sci & Design Thinking Processes (CS&DTP)	<ol style="list-style-type: none"> 1. Fostering an Inclusive Computing and Design Culture 2. Collaborating Around Computing and Design 3. Recognizing and Defining Computational Problems 4. Developing and Using Abstractions 5. Creating Computational Artifacts 6. Testing and Refining Computational Artifacts 7. Communicating About Computing and Design
Standards: From 2020 New Jersey Student Learning Standards – Computer Science and Design Thinking	
Computing Systems(CS):	<ul style="list-style-type: none"> • CS.8.1 - The study of human & computer interaction can improve the design of devices and extend the abilities of humans • CS.8.3 - Troubleshooting a problem is more effective when knowledge of the specific device along with a systematic process is used to identify the source of a problem. • CS.12.2 - A computing system involves interaction among the user, hardware, application software and system software. • CS.12.3 - Successful troubleshooting of complex problems involves multiple approaches including research, analysis, reflection, interaction with peers and drawing on past experiences.
Impact of Computing(IC):	<ul style="list-style-type: none"> • IC.8.1 - Advancements in computing technology can change individuals' behaviors • IC.8.2 - Society is faced with trade offs due to the increasing globalization and automation that computing brings • IC.12.1 - The design and use of computing technologies and artifacts can positively or negatively affect equitable access to information and opportunities.
Data and Analysis(DA):	<ul style="list-style-type: none"> • DA.8.1 - People use digital devices and tools to automate the collection, use, and transformation of data. • DA.8.3 - Data is represented in many formats. Software tools translate the low-level representation of bits into a form understandable by individuals. Data is organized and accessible based on the application used to store it. • DA.12.1 - Individuals select digital tools and design automated processes to collect, transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information. • DA.12.3 - Large data sets can be transformed, generalized, simplified, and presented in different ways to influence how individuals interpret and understand the underlying information.
Algorithms and Programming (AP):	<ul style="list-style-type: none"> • AP.8.1 - Individuals design algorithms that are reusable in many situations • AP.8.2 - Algorithms that are readable are easier to follow, test, and debug • AP.8.3 - Programmers create variables to store data values of different types and perform appropriate operations on their values. • AP.8.4 - Control structures are selected and combined in programs to solve more complex problems • AP.8.5 - Programs use procedures to organize code and hide implementation details. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability. • AP.12.1 - Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation • AP.12.2 - Programmers choose data structures to manage program complexity based on functionality, storage, and performance trade-offs • AP.12.3 - Trade-offs related to implementation, readability, and program performance are considered when selecting and combining control structures. • AP.12.4 - Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks.
Networks and the Internet: (NI)	<ul style="list-style-type: none"> • NI.8.2 - The information sent and received across networks can be protected from unauthorized access and modification in a variety of ways • NI.12.1 - The scalability and reliability of the internet are enabled by the hierarchy and redundancy of networks

Suggested Time Schedule

Unit Number	Unit Name	Total Days (est)
1	Intro to Computers	15
2	Basics of Programming	25
3	Control Structures I - Conditional Statements	20
4	Control Structures II - Loops	25
5	Functions	15
6	Modules	20
7	Data Structures & Statistics	35
Total		155

Unit of Study: 1. Introduction to Computers**Targeted State Standard(s):**

- CS.8.1 - The study of human & computer interaction can improve the design of devices and extend the abilities of humans
- CS.8.3 - Troubleshooting a problem is more effective when knowledge of the specific device along with a systematic process is used to identify the source of a problem.
- CS.12.2 - A computing system involves interaction among the user, hardware, application software and system software.
- CS.12.3 - Successful troubleshooting of complex problems involves multiple approaches including research, analysis, reflection, interaction with peers and drawing on past experiences.
- IC.8.1 - Advancements in computing technology can change individuals' behaviors
- IC.8.2 - Society is faced with trade offs due to the increasing globalization and automation that computing brings
- IC.12.1 - The design and use of computing technologies and artifacts can positively or negatively affect equitable access to information and opportunities
- NI.8.2 - The information sent and received across networks can be protected from unauthorized access and modification in a variety of ways
- NI.12.1 - The scalability and reliability of the internet are enabled by the hierarchy and redundancy of networks
- AP.8.1 - Individuals design algorithms that are reusable in many situations
- AP.8.2 - Algorithms that are readable are easier to follow, test, and debug

Unit Objectives/Enduring Understandings:

- Students will understand the origins of computing, its strengths and weaknesses, and the relationship between hardware, software, and user.

Essential Questions:

- What effects have the increased prevalence of computers, globalization, and automation had on our world/society?
- What are the 4 essential components of every computing device?
- What are the responsibilities of the programmer in the computer application design process?

Unit Assessment:

- Computing Impact Project - Research of a Computing Technology and its impact on people's lives (positive and negative).
- Computer Pioneer Poster Project - Make a biographical poster of a famous computing pioneer

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts (What students will know)	Skills (What students will be able to do)	Activities/ Strategies (Technology Implementation/ Interdisciplinary Connections)	Assessment Check Points
<ul style="list-style-type: none">• Identify the differences between hardware and software and the purpose of each.• Identify the 4 main components of every computing device.• Identify the impact of computing on society• Convert numbers between number systems (binary, decimal, hexadecimal)	<ul style="list-style-type: none">• History of Computers• Parts of a Computer• Impacts of Computing• The Internet and Digital Data• Digital Literacy & Organization• Problem Solving & Critical Thinking• Number Systems	<ul style="list-style-type: none">• Identify the components of a computer and the difference between hardware and software.• Identifying the impacts that computers have on society.• Identifying how computers interpret, translate, and exchange data.• Creating a program in the IDE• Maintaining an organized digital workspace.• Convert between number systems	<ul style="list-style-type: none">• Powerpoint/Slides presentations• Research Projects• Creating Computational Artifacts	<ul style="list-style-type: none">• Reflection questions• Show What You Know Questions.• Workspace check
Resources: Essential Materials, Supplementary Materials, Links to Best Practices <ul style="list-style-type: none">• Textbook and related resources• MacBook• PyCharm Edu• Website articles• Video Library			Instructional Adjustment: Modifications, student difficulties, possible misunderstandings <ul style="list-style-type: none">• Circulate during work time to answer questions and provide clarity.• Utilize pair programming to ensure students have a person with whom to collaborate and discuss difficulties.(CS&DTP - 1, 2, & 7)	

Unit of Study: 2. Basics of Programming**Targeted State Standard(s):**

- AP.8.2 - Algorithms that are readable are easier to follow, test, and debug
- AP.8.3 - Programmers create variables to store data values of different types and perform appropriate operations on their values.

Unit Objectives/Enduring Understandings:

- Making programs interactive by using input and output
- Using variables to make program functional
- Use math operators and functions to perform calculations.
- Strings store an array of characters

Essential Questions:

- How do we define and use variables?
- How do we take input from the user?
- What is the purpose of an escape sequence?
- How do we format numbers when printing?
- Why are comments and documentation important?

Unit Assessment:

- Programming Project using output, input, variables, math methods, and string formatting for decimals

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts (What students will know)	Skills (What students will be able to do)	Activities/ Strategies (Technology Implementation/ Interdisciplinary Connections)	Assessment Check Points
<ul style="list-style-type: none">Using output to the console to communicate with the user of the program.Using variables to take and store dataUsing math operators and functions to perform calculations.Taking input from the user for use in the program.Using the random module to generate random numbersGetting the current date and time from the computer	<ul style="list-style-type: none">Comments & Program DocumentationOutput & StringsVariables & StorageMath Operators and FunctionsString FormattingRandom NumbersUser InputDate/Time Object	<ul style="list-style-type: none">Write a single line commentWrite a multi line commentWrite the standard program documentation at the beginning of every programPrint to the consoleUse String methods to manipulate stringsUse the math operators and functions to perform calculationsFormat strings for decimal outputGenerate a random number in a given rangeTake input from the user.Get current date/time from the computer	<ul style="list-style-type: none">Powerpoint/Slides presentationsDiscovery ActivitiesMath/Science FormulasCollaborative/Pair Programming (12.CS.3,8.DA.5)Passion Projects(12.CS.3, 8.DA.5, 8.AP.5, 12.ED.1)	<ul style="list-style-type: none">Reflection QuestionsShow What You Know QuestionsClasswork programming assignmentsWorkspace check
Resources: Essential Materials, Supplementary Materials, Links to Best Practices <ul style="list-style-type: none">Textbook and related resourcesMacBookPyCharm EduWebsite articlesVideo Library			Instructional Adjustment: Modifications, student difficulties, possible misunderstandings <ul style="list-style-type: none">Circulate during work time to answer questions and provide clarity.Utilize pair programming to ensure students have a person with whom to collaborate and discuss difficulties.(CS&DTP - 1, 2, & 7)	

Unit of Study: 3. Conditional Statements**Targeted State Standard(s):**

- AP.8.1 - Individuals design algorithms that are reusable in many situations
- AP.8.2 - Algorithms that are readable are easier to follow, test, and debug
- AP.8.4 - Control structures are selected and combined in programs to solve more complex problems
- AP.12.3 - Trade-offs related to implementation, readability, and program performance are considered when selecting and combining control structures.

Unit Objectives/Enduring Understandings:

- Format of an if/else and elif statement.
- Proper indentation is needed to indicate the body of a conditional statement
- Nesting conditional statements allows for more complex tasks to be completed.

Essential Questions:

- When should we use the if/else and elif statement?
- Why is it important to use proper indentation when writing if/else and elif statements?
- How do we represent an if/else/elif statement on a flowchart algorithm?

Unit Assessment:

- Programming assignment using if/else/elif statements and nesting.

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts (What students will know)	Skills (What students will be able to do)	Activities/ Strategies (Technology Implementation/ Interdisciplinary Connections)	Assessment Check Points
<ul style="list-style-type: none">• Determine the true/false outcome of a boolean statement.• Use the if statement to skip over code if a condition is not met.• Use the else statement as a default to offer a second choice• Use the elif statement to expand an if/else statement to model more outcomes.• Nest block statements inside of each other.	<ul style="list-style-type: none">• Boolean Logic• If Statements• Else Statement• Nesting If/Else Statements• Elif Statements• Flowchart Algorithms	<ul style="list-style-type: none">• Evaluate boolean statements.• Write if/else and elif statements• Nest if/else statements• Read and write flowchart algorithms	<ul style="list-style-type: none">• Powerpoint/Slides presentations• Discovery Activities• Math/Science Formulas• Collaborative/Pair Programming (12.CS.3,8.DA.5)• Passion Projects(12.CS.3, 8.DA.5, 8.AP.5, 12.ED.1)	<ul style="list-style-type: none">• Reflection Questions• Show What You Know Questions• Classwork programming assignments• Workspace check
Resources: Essential Materials, Supplementary Materials, Links to Best Practices <ul style="list-style-type: none">• Textbook and related resources• MacBook• PyCharm Edu• Website articles• Video Library			Instructional Adjustment: Modifications, student difficulties, possible misunderstandings <ul style="list-style-type: none">• Circulate during work time to answer questions and provide clarity.• Utilize pair programming to ensure students have a person with whom to collaborate and discuss difficulties.(CS&DTP - 1, 2, & 7)	

Unit of Study: 4. Loops**Targeted State Standard(s):**

- AP.8.1 - Individuals design algorithms that are reusable in many situations
- AP.8.2 - Algorithms that are readable are easier to follow, test, and debug
- AP.8.4 - Control structures are selected and combined in programs to solve more complex problems
- AP.12.3 - Trade-offs related to implementation, readability, and program performance are considered when selecting and combining control structures.

Unit Objectives/Enduring Understandings:

- Format of a For Loop and While Loop
- How to use a loop to repeat code
- How to represent a loop on a flowchart algorithm
- How to nest loops and other structures.

Essential Questions:

- What is the difference between a for loop and a while loop?
- When would you use a for loop and when would you use a while loop?

Unit Assessment:

- Programming Project that uses both for and while loops.

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts (What students will know)	Skills (What students will be able to do)	Activities/ Strategies (Technology Implementation/ Interdisciplinary Connections)	Assessment Check Points
<ul style="list-style-type: none"> • Using a for loop over a range to repeat a fixed number of times. • Using a while loop in a fixed looping setting. • Using a while loop in a variable looping setting. 	<ul style="list-style-type: none"> • For loop over a range • Fixed length application of while loops • Variable length application length of while loops • Nesting Loops 	<ul style="list-style-type: none"> • Use a For Loop to repeat code • Use a While Loop to repeat code • Nest Loops to solve more complex tasks. 	<ul style="list-style-type: none"> • Powerpoint/Slides presentations • Discovery Activities • Math/Science Formulas • Collaborative/Pair Programming (12.CS.3,8.DA.5) • Passion Projects(12.CS.3, 8.DA.5, 8.AP.5, 12.ED.1) 	<ul style="list-style-type: none"> • Reflection Questions • Show What You Know Questions • Classwork programming assignments • Workspace check
Resources: Essential Materials, Supplementary Materials, Links to Best Practices <ul style="list-style-type: none"> • Textbook and related resources • MacBook • PyCharm Edu • Website articles • Video Library 			Instructional Adjustment: Modifications, student difficulties, possible misunderstandings <ul style="list-style-type: none"> • Circulate during work time to answer questions and provide clarity. • Utilize pair programming to ensure students have a person with whom to collaborate and discuss difficulties.(CS&DTP - 1, 2, & 7) 	

Unit of Study: 5. User Defined Functions**Targeted State Standard(s):**

- AP.8.1 - Individuals design algorithms that are reusable in many situations
- AP.8.2 - Algorithms that are readable are easier to follow, test, and debug
- AP.8.5 - Programs use procedures to organize code and hide implementation details. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.
- AP.12.1 - Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation

Unit Objectives/Enduring Understandings:

- We use functions to simplify our programs and to abstract away details.

Essential Questions:

- How do we define a function?
- Where do we define a function?
- What do we do to return a value?
- How do we activate a function?
- What is the difference between a function and a lambda?

Unit Assessment:

- Programming Assignment that combines functions and all previous content.

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts (What students will know)	Skills (What students will be able to do)	Activities/ Strategies (Technology Implementation/ Interdisciplinary Connections)	Assessment Check Points
<ul style="list-style-type: none"> • Writing a function and calling it • Writing and using a lambda 	<ul style="list-style-type: none"> • User Defined Functions • Lambdas 	<ul style="list-style-type: none"> • Define a function • Call a function • Return a value • Write a lambda 	<ul style="list-style-type: none"> • Powerpoint/Slides presentations • Discovery Activities • Math/Science Formulas • Collaborative/Pair Programming (12.CS.3,8.DA.5) • Passion Projects(12.CS.3, 8.DA.5, 8.AP.5, 12.ED.1) 	<ul style="list-style-type: none"> • Reflection Questions • Show What You Know Questions • Classwork programming assignments • Workspace check
Resources: Essential Materials, Supplementary Materials, Links to Best Practices <ul style="list-style-type: none"> • Textbook and related resources • MacBook • PyCharm Edu • Website articles • Video Library 			Instructional Adjustment: Modifications, student difficulties, possible misunderstandings <ul style="list-style-type: none"> • Circulate during work time to answer questions and provide clarity. • Utilize pair programming to ensure students have a person with whom to collaborate and discuss difficulties.(CS&DTP - 1, 2, & 7) 	

Unit of Study: 6. Modules**Targeted State Standard(s):**

- AP.12.4 - Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks.

Unit Objectives/Enduring Understandings:

- Modules can be used to simplify code and offer abstraction.
- Modules can be designed to perform tasks and shared with others to aid them in solving problems

Essential Questions:

- What is abstraction and how does it help programmers write programs?
- How do we create a module?
- How do we import a module?
- How do we access variables/functions from a module?

Unit Assessment:

- Programming Project incorporating previous topics

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts (What students will know)	Skills (What students will be able to do)	Activities/ Strategies (Technology Implementation/ Interdisciplinary Connections)	Assessment Check Points
<ul style="list-style-type: none"> • Creating a module • Importing a module • Working with the variables or functions from a module 	<ul style="list-style-type: none"> • Modules in Python 	<ul style="list-style-type: none"> • Create a module • Import a module or part of a module • Rename a module in a program 	<ul style="list-style-type: none"> • Powerpoint/Slides presentations • Discovery Activities • Math/Science Formulas • Collaborative/Pair Programming (12.CS.3,8.DA.5) • Passion Projects(12.CS.3, 8.DA.5, 8.AP.5, 12.ED.1) 	<ul style="list-style-type: none"> • Reflection Questions • Show What You Know Questions • Classwork programming assignments • Workspace check
Resources: Essential Materials, Supplementary Materials, Links to Best Practices <ul style="list-style-type: none"> • Textbook and related resources • MacBook • PyCharm Edu • Website articles • Video Library 			Instructional Adjustment: Modifications, student difficulties, possible misunderstandings <ul style="list-style-type: none"> • Circulate during work time to answer questions and provide clarity. • Utilize pair programming to ensure students have a person with whom to collaborate and discuss difficulties.(CS&DTP - 1, 2, & 7) 	

Unit of Study: 7. Data Structures & Statistics**Targeted State Standard(s):**

- AP.12.2 - Programmers choose data structures to manage program complexity based on functionality, storage, and performance trade-offs
- DA.8.1 - People use digital devices and tools to automate the collection, use, and transformation of data.
- DA.12.1 - Individuals select digital tools and design automated processes to collect, transform, generalize, simplify, and present large data sets in different ways to influence how other people interpret and understand the underlying information.
- DA.12.3 - Large data sets can be transformed, generalized, simplified, and presented in different ways to influence how individuals interpret and understand the underlying information.

Unit Objectives/Enduring Understandings:

- Python uses 4 data structures
- Each data structure has its strengths and weaknesses.
- When to use each data structure
- Using data structures to analyze statistics

Essential Questions:

- Elaborate on the differences between lists, tuples, sets, and dictionaries.
- When should you choose to use a list/set/tuple/dictionary?

Unit Assessment:

- Programming Project involving prior content and the use of statistics.

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	Concepts (What students will know)	Skills (What students will be able to do)	Activities/ Strategies (Technology Implementation/ Interdisciplinary Connections)	Assessment Check Points
<ul style="list-style-type: none">• Declare and initialize a:<ul style="list-style-type: none">◦ List◦ Tuple◦ Set◦ Dictionary• Explain the differences between list, tuple, set, dictionary• Evaluate a dataset and find mean, median, mode, standard deviation, and variance• Create a graph of a dataset	<ul style="list-style-type: none">• Data Structures:<ul style="list-style-type: none">◦ List◦ Tuple◦ Set◦ Dictionary• Use the mean, median, and mode methods of the numpy module• Use the standard deviation and variance methods of the numpy module• Use the percentile method of the numpy module• Use the matplotlib library to create and show graphs from datasets.	<ul style="list-style-type: none">• Use data structures to store multiple pieces of information• Analyze data in a dataset using the numpy module• Use the matplotlib library to create graphs of a dataset	<ul style="list-style-type: none">• Powerpoint/Slides presentations• Discovery Activities• Math/Science Formulas• Collaborative/Pair Programming (12.CS.3,8.DA.5)• Passion Projects(12.CS.3, 8.DA.5, 8.AP.5, 12.ED.1)	<ul style="list-style-type: none">• Reflection Questions• Show What You Know Questions• Classwork programming assignments• Workspace check
Resources: Essential Materials, Supplementary Materials, Links to Best Practices <ul style="list-style-type: none">• Textbook and related resources• MacBook• PyCharm Edu• Website articles• Video Library			Instructional Adjustment: Modifications, student difficulties, possible misunderstandings <ul style="list-style-type: none">• Circulate during work time to answer questions and provide clarity.• Utilize pair programming to ensure students have a person with whom to collaborate and discuss difficulties.(CS&DTP - 1, 2, & 7)	