SCHOOL DISTRICT OF THE CHATHAMS
CURRICULUM PROFILE

CONTENT AREA(S): Design & Technology **GRADE LEVEL(S):** 9-12

COURSE: Game Design w/ Python **TIME FRAME:** Semester (2.5 Credits)

PREREQUISITES: None

I. 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 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.

II. Units of Study

Unit 1: Karel the Robot
Unit 2: Extending the Robot Programming Language
Unit 3: Conditionally Executing Instructions
Unit 4: Computer Architecture
Unit 5: Variables and Strings
Unit 6: Python and PyGames

III. Essential Questions *(The open-ended, provocative questions that help frame inquiry)*

Unit 1: Introduction to Python using Karel the Robot

- 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

- 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?
- (Project) What impact have computers had on history?

Unit 3: Conditionally Executing Instructions

- 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?
- How do we write our own boolean methods?

Unit 4: Computer Architecture

- What are the hardware components of a computer and their functions?
- What are the software components of a computer and their functions?
- What are number systems? (Binary)
- How do analog and digital compare?
- What are networks?
- What is the internet?

Unit 5: Variables and Strings

- 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 6: Final Projects

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

IV. Learning Objectives

Objectives align with the New Jersey Student Learning Standards (NJSLS) Standard 8.2 - Strand E. It also incorporates concepts of problem solving in Mathematics and the content areas of Technological Literacy and 21st Century Life and Careers.

V. Instructional Materials

Monty Karel, by Joseph Bergin

Supplementary Materials

- Online documentation for the Python programming language
- Online documentation for the IDE
- Software for development for games (PyGames, etc)
- Teacher Generated Materials

Revision Date: 6/2018

VI. Key Performance and Benchmark Tasks

Unit 1: Introduction to Python using Karel the Robot

- 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.

Unit 2: Extending the Robot Programming Language

- 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 an understanding of the progression of computers, from the mechanical to the electronic, as well as notable recent events such as the dot com boom and y2k.

Unit 3: Conditionally Executing Instructions

- 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

Unit 4: Computer Architecture

- Demonstrate an understanding of basic computer architecture (hardware/software).
- Demonstrate an understanding of how number systems work.
- Demonstrate a basic understanding of networks and the internet.
- Create a program that prints to the console

Unit 5: Variables and Strings

- 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.

Unit 6: Final Projects

- Design and implement a Python game referencing the techniques learned through the semester

A major focus of this class is developing problem solving skills. In an effort to build those skills, the instruction into programming in Python is infused with various problem solving activities. Generally given once a week, students explore different problems designed to promote lateral thinking and develop their ability to analyze problems and revise solutions.