I.   CATALOG INFORMATION

A.   Discipline: COMPUTER SCIENCE (CS)

B.   Subject Code and Number: CS M10P

C.   Course Title: Introduction to Computer Programming Using Python Language

D.   Credit Course units:

Units: 4

Lecture Hours per week: 3

Lab Hours per week : 3

Variable Units : No

E.   Student Learning Hours:

Lecture Hours:

Classroom hours: 52.5 - 52.5

Laboratory/Activity Hours:

Laboratory/Activity Hours 52.5 - 52.5

Total Combined Hours in a 17.5 week term: 105 - 105

F.   Non-Credit Course hours per week _____

G.   May be taken a total of: [X] 1 [ ] 2 [ ] 3 [ ] 4 time(s) for credit

H.   Is the course co-designated (same as) another course: No [X] Yes [ ]
If YES, designate course Subject Code & Number: _____

I.   Course Description:

Provides an introduction to computer programming and algorithm design using the Python programming language. Covers the fundamentals of computer programming: basic data types, switching and looping constructs, functions, recursion, objects, arrays, and lists, and basic input and output, both interactive and with files. Explains some principles of algorithm design and analysis as well as techniques for testing programs.

J.   Entrance Skills

*Prerequisite: No [X] Yes [ ] Course(s)
_____

*Corequisite: No [X] Yes [ ] Course(s)
_____

Limitation on Enrollment: No [X] Yes [ ]
_____

Recommended Preparation: No [X] Yes [ ] Course(s)
_____

Other: No [X] Yes [ ]

K.   Other Catalog Information:

II.   COURSE OBJECTIVES

Upon successful completion of the course, a student will be able to:

		Methods of evaluation will be consistent with, but not limited by, the following types or examples.
1	explain computer programming concepts.	Graded assignments Programming projects Quizzes Exams
2	convert algorithms to Python programs.	Graded assignments Programming projects Quizzes Exams Laboratory reports
3	design modular Python programs using functions.	Graded assignments Programming projects Quizzes Exams Laboratory reports
4	design programs with Interactive Input and Output.	Graded assignments Programming projects Quizzes Exams Laboratory reports
5	design programs utilizing arithmetic expressions.	Graded assignments Programming projects Quizzes Exams
6	design programs utilizing repetition.	Graded assignments Programming projects Quizzes Exams
7	design programs utilizing decision-making.	Graded assignments Programming projects Quizzes Exams

8	design programs utilizing arrays; develop simple search and sort algorithms using arrays.	Graded assignments Programming projects Quizzes Exams
9	design programs using file Input and Output.	Graded assignments Programming projects Quizzes Exams Laboratory reports
10	design object-oriented programs that include data (private data members) and code (functions) together which is called encapsulation.	Graded assignments Programming projects Quizzes Exams

III.   COURSE CONTENT

Estimated %	Topic	Learning Outcomes
Lecture (must total 100%)		

8.00%	Fundamentals of the Python language for computer science programming
	  A. Use of the computer and computer languages
	  B. Problem analysis
	  C. Meaning of an algorithm
		1

9.00%	Elementary data types and operations
	  A. Data types
	    1. Numeric
	    2. String
	  B. Operations
	    1. Assignment operators
	    2. Arithmetical operations
		1, 2, 5

16.00%	Design with control structures
	  A. Design with decision-making steps: use of if, if else, nested if, multi alternative if, and switch statements
	  B. Design with repetitions steps: use of iteration statements
	  C. Pretest and post-test loops and software situations identifying their use
	  D. Introduction to End-of-file control loops and file processing based on them
		1, 2, 6, 7

22.00%	Design with sub programs (blockstructured programming style)
	  A. Functions with and without parameters/return values
	  B. Use of control structures in the context of sub programs
	  C. Keyword arguments and argument unpacking
	  D. Built-in functions in Python
	  E. Text processing with built-in and user-defined functions in Python
	  F. Functions returning multiple parameters
	  G. Functions returning tuples
		1, 2, 3

	Errors and exceptions	

7.00%	A. General introductions to errors and exceptions B. Raising and handling exceptions C. User-defined exceptions	2, 4, 9

14.00%	Input/Output file manipulations
	  A. Use of text input files
	  B. Creating output files
	  C. Output formatting
	  D. Methods of file object
		2, 9

18.00%	Design with large block of data in main memory
	  A. Use of structured data types:
	    1. Arrays
	    2. Structures
	    3. Lists
	    4. Tuples
	  B. Operations
	    1. Read
	    2. Print
	    3. Search
	    4. Sort
		1, 8

6.00%	Classes and objects
	  A. Demonstrating design and development
	  B. Use of a user-defined class
		2, 10

Lab (must total 100%)		

7.00%	Fundamentals of the Python language for computer science programming
	  A. Use of the computer and computer languages
	  B. Problem analysis
	  C. Meaning of an algorithm
		1, 2

8.00%	Elementary data types and operations
	  A. Data types
	    1. Numeric
	    2. String
	  B. Operations
	    1. Assignment operators
	    2. Arithmetical operations
		2

16.50%	Design with control structures
	  A. Design with decision-making steps: use of if, if-else, nested if, multi alternative if, and switch statements
	  B. Design with repetitions steps: use of iteration statements
	  C. Pretest and post-test loops and software situations identifying their use
	  D. Introduction to End-of-file control loops and file processing based on them
		1, 2, 3, 4, 5, 6, 7, 8, 9, 10

22.00%	Design with sub programs (block structured programming style)
	  A. Functions with and without parameters/return values
	  B. Use of control structures in the context of sub programs
	  C. Keyword arguments and argument unpacking
	  D. Built-in functions in Python
	  E. Text processing with built-in and user-defined functions in Python
	  F. Functions returning multiple parameters
	  G. Functions returning tuples
		4, 5

6.50%	Errors and exceptions
	  A. General introductions to errors and exceptions
	  B. Raising and handling exceptions
		7

	C. User-defined exceptions	

14.00%	Input/Output File manipulations
	  A. Use of text input files
	  B. Creating output files
	  C. Output formatting
	  D. Methods of file object
		5

20.00%	Design with large block of data in main memory
	  A.Use of structured data types:
	    1. Arrays
	    2. Structures
	    3. Lists
	    4. Tuples
	  B. Operations
	    1. Read
	    2. Print
	    3. Search
	    4. Sort
		8

6.00%	Classes and objects
	  A. Demonstrating design and development
	  B. Use of a user- defined class
		1, 2, 3, 4,
		5, 6, 7, 8,
		9, 10

IV.   TYPICAL ASSIGNMENTS

A.   Writing assignments

	Writing assignments are required. Possible assignments may include, but are not limited to:
1	write a Python program without a graphical user interface that calculates and displays the mortgage payment amount given the amount of the mortgage, the term of the mortgage, and the interest rate of the mortgage by the user.
2	write a Python program that asks the user to enter the monthly costs for the following expenses incurred from operating his or her automobile: loan payment, insurance, gas, oil, tires, and maintenance. The program should then pass these values to a function to calculate the total monthly cost of these expenses, the total annual cost of these expenses, and display them.
3	write a Python program that reads the contents of the CorrectAnswers.txt file into a char array, and then reads the contents of another file, containing a student's answers, into a second char array. The program should determine the number of questions that the student missed.

B.   Appropriate outside assignments

	Appropriate outside assignments are required. Possible assignments may include, but are not limited to:
1	analyze the strengths and weaknesses of educational software programs.
2	design a new web app using Python to find local software companies.
3	work in groups to write a program that calculates a customer's monthly bill. It should ask which package the customer has purchased and how many minutes were used. It should then display the total amount due.

C.   Critical thinking assignments

	Critical thinking assignments are required. Possible assignments may include, but are not limited to:
	1- In this assignment you will write two separate programs, so you must create two projects. Both will do the same thing, but they will use different tools. Both programs will request the user to input a line of text. The program must then

1	analyze the text to determine whether or not it is a palindrome. A palindrome reads the same forwards and backwards while ignoring punctuation and spaces. For example, these are all palindromes: abba abccccba radar Dad Madam, I'm Adam A man, a plan, a canal, Panama!!! The letter a is also a palindrome, as is any letter by itself. Finally, using this definition, the empty string is also considered a palindrome. A palindrome is NOT case sensitive. The expression doesn't even need not make any sense, such as with abccccba. In the first project, you are restricted to using the "string" libraryyou may not use "cstring". In the second project, the situation is reversed? you may only use "cstring", not "string". No global variables, global constants or global arrays are allowed. (The one exception is that I will allow a global constant for the maximum size of the array, which I have a set at 50). For both programs, you must create a loop that allows the user to test several expressions until the user chooses to quit. However, it must allow for any kind of user response such as y, Y, yes or Yes for the response. Functions are required. One function should test an expression to see if it is a palindromeit should have the prototypes: bool isPal(const string&) and bool isPal(const char[]), depending on which program you are doing. You should also have a function that removes punctuation from the expression how you code this is up to you. I will give suggestions each day in class. After finishing both programs, compare and contrast the two approaches in a short, but well written paragraph.
2	2- Analyze and interpret the relation between number of hours study and the test result for students at college. Collect your data from the library for different typical courses. We want to examine the relationship between amount of time spent studying for exam (X) in hours and the score that person makes on exam (Y). Use Python programming to calculate coefficient of correlation and interpret it. a. Determine the regression equation. b. Determine the value of Y' when X is 9 (if you study 9 hours, what is your grade?). Here is a sample data: X Y 2 65 3 70 3 75 4 70 5 85 6 85 7 90
	3- The city council of Pine Bluffs is considering increasing the number of police in an effort to reduce crime. Before making a final decision, the council asks the Chief of Police to survey other cities of similar size to determine the relationship between the number of police and the number of crimes reported. The Chief gathered the following sample information. City Oxford, Number of police=15, Number of Crimes= 17 City Starkville, Number of police=17, Number of Crimes=13 City Danville, Number of police=25, Number of Crimes=5

3	City Athens, Number of police=27, Number of Crimes=7
	City Holgate, Number of police= 17, Number of Crimes=7
	City Carey, Number of police=12, Number of Crimes= 21
	City Whistler, Number of police=11, Number of Crimes=19
	City Woodville, Number of police=22, Number of Crimes=6
	a. Design and Determine the coefficient of correlation using Python and Pearson formula. b. Analyze and Interpret this statistical measures. Does it surprise you that the relationship is inverse?

V. METHODS OF INSTRUCTION

Methods of instruction may include, but are not limited to:

[ ] Distance Education – When any portion of class contact hours is replaced by distance education delivery mode (Complete DE Addendum, Section XV)

[X] Lecture/Discussion

[X] Laboratory/Activity

[X] Other (Specify)
Bring a guest speaker from industry to practice real world problems to develop proper programming skills and techniques
Student/Instructor interaction using questions and answers
Projects and/or group work to enhance student understanding of the concepts

[ ] Optional Field Trips

[ ] Required Field Trips

VI. METHODS OF EVALUATION
Methods of evaluation may include, but are not limited to:

[X]	Essay Exam	[X]	Classroom Discussion	[X]	Skill Demonstration
[X]	Problem Solving Exam	[X]	Reports/Papers/ Journals	[X]	Participation
[X]	Objective Exams	[X]	Projects	[X]	Other (specify)

Weekly computer lab assignments

VII. REPRESENTATIVE TEXTS AND OTHER COURSE MATERIALS

Gaddis Tony. Starting Out with Python. 4th ed. Pearson, 2017.

Guzdial, Mark, and Barbara Ericson. Introduction to Computing and Programming in Python. 4th ed. Pearson, 2016.

<u>Python 3.5.2</u>. All Python releases are Open Source, 3.5.2 ed.
Python Software Foundation

Python Programming: An Introduction to Computer Science, 3rd Ed.
by John Zelle, ISBN-13: 978-1590282755, ISBN-10: 1590282752

VIII. STUDENT MATERIALS FEES

$\boxed{X}$ No $\square$ Yes

IX. PARALLEL COURSES

College	Course Number	Course Title	Units
Humboldt State Unv.	CS 232	Python Programming	3
UC Santa Cruz	CMPS 5P	Introduction to Programming in Python	5
Santa Monica College	CS 87A	Python Programming	3
CSU Fullerton	CPSC 223P	Introduction to Python Programming	3
Mission College	CIS 7	Python Programming	4
Irvine Valley College	CS 10	Introduction to Programming Using Python	3

X. MINIMUM QUALIFICATIONS

Courses Requiring a Masters Degree:
Master's degree in computer science or computer engineering OR bachelor's degree in either of the above AND master's degree in mathematics, cybernetics, business administration, accounting or engineering OR bachelor's degree in engineering AND master's degree in cybernetics, engineering, mathematics, or business administration OR bachelor's degree in mathematics AND master's degree in cybernetics, engineering mathematics, or business administration OR bachelor's degree in any of the above AND a master's degree in information science, computer information systems, or information systems OR the equivalent. (NOTE: Courses in the use of computer programs for application to a particular discipline may be classified, for minimum qualifications purposes, under the discipline of the application.)

XI. ARTICULATION INFORMATION

A. Title V Course Classification:

1. This course is designed to be taken either:

$\square$ Pass/No Pass only (no letter grade possible); or

$\boxed{X}$ Letter grade (P/NP possible at student option)

2. Degree status:
Either $\boxed{X}$ Associate Degree Applicable; or $\square$ Non-associate Degree Applicable

B. Moorpark College General Education:

1. Do you recommend this course for inclusion on the Associate Degree General Education list?
Yes: $\square$ No: $\boxed{X}$ If YES, what section(s)?

$\square$ A1 - Natural Sciences - Biological Science

$\square$ A2 - Natural Sciences - Physical Science

$\square$ B1 - Social and Behavioral Sciences - American History/Institutions

$\square$ B2 - Social and Behavioral Sciences - Other Social Behavioral Science

$\square$

☐ C1 - Humanities - Fine or Performing Arts

☐ C2 - Humanities - Other Humanities

☐ D1 - Language and Rationality - English Composition

☐ D2 - Language and Rationality - Communication and Analytical Thinking

☐ E1 - Health/Physical Education

☐ E2 - PE or Dance

☐ F - Ethnic/Gender Studies

C. California State University(CSU) Articulation:

1. Do you recommend this course for transfer credit to CSU?   Yes: ☒ No: ☐

2. If YES do you recommend this course for inclusion on the CSU General Education list?
   Yes: ☐ No: ☒ If YES, which area(s)?

   A1 ☐   A2 ☐   A3 ☐   B1 ☐   B2 ☐   B3 ☐   B4 ☐

   C1 ☐   C2 ☐   D1 ☐   D2 ☐   D3 ☐   D4 ☐   D5 ☐

   D6 ☐   D7 ☐   D8 ☐   D9 ☐   D10 ☐   E ☐

D. University of California (UC) Articulation:

1. Do you recommend this course for transfer to the UC?   Yes: ☒ No: ☐

2. If YES do you recommend this course for the Intersegmental General Education Transfer Curriculum (IGETC)?   Yes: ☐ No: ☒

   IGETC Area 1: English Communication

   ☐ English Composition

   ☐ Critical Thinking-English Composition

   ☐ Oral Communication

   IGETC Area 2: Mathematical Concepts and Quantitative Reasoning

   ☐ Mathematical Concepts

   IGETC Area 3: Arts and Humanities

   ☐ Arts

   ☐ Humanities

   IGETC Area 4: Social and Behavioral Sciences

   ☐ Anthropology and Archaeology

   ☐ Economics

   ☐ Ethnic Studies

   ☐ Gender Studies

   ☐

☐ Geography

☐ History

☐ Interdisciplinary, Social & Behavioral Sciences

☐ Political Science, Government & Legal Institutions

☐ Psychology

☐ Sociology & Criminology

IGETC Area 5: Physical and Biological Sciences (mark all that apply)

☐ Physical Science Lab or Physical Science Lab only (none-sequence)

☐ Physical Science Lecture only (non-sequence)

☐ Biological Science

☐ Physical Science Courses

☐ Physical Science Lab or Biological Science Lab Only (non-sequence)

☐ Biological Science Courses

☐ Biological Science Lab course

☐ First Science course in a Special sequence

☐ Second Science course in a Special Sequence

☐ Laboratory Activity

☐ Physical Sciences

IGETC Area 6: Language other than English

☐ Languages other than English (UC Requirement Only)

☐ U.S. History, Constitution, and American Ideals (CSU Requirement ONLY)

☐ U.S. History, Constitution, and American Ideals (CSU Requirement ONLY)

XII. REVIEW OF LIBRARY RESOURCES

A. What planned assignment(s) will require library resources and use?

The following assignments require library resources:
Use of the Library's print and online resources to locate current articles in newspapers, magazines and scholarly journals in preparation for a report on various aspects of code and coding practices.

B. Are the currently held library resources sufficient to support the course assignment?

YES: ☒ NO: ☐

If NO, please list additional library resources needed to support this course.

XIII. PREREQUISITE AND/OR COREQUISITE JUSTIFICATION

CS M10P: Not Applicable

XIV. WORKPLACE PREPARATION

Required for career technical courses only. A career technical course/program is one with the primary goal to prepare students for employment immediately upon course/program completion, and/or upgrading employment skills.

Detail how the course meets the Secretary of Labors Commission on the Achievement of Necessary Skills (SCANS) areas. (For a description of the competencies and skills with a listing of what students should be able to do, go to: http://www.ncrel.org/sdrs/areas/issues/methods/assment/as7scans.htm)

The course will address the SCANS competency areas:

1. Resources: the students will learn to use in an effective and efficient manner the various resources at their disposal to design and analyze an application program for a variety of businesses. Resources include but are not limited to technological resources, facilities resources, economical resources, and human resources.

2. Interpersonal: the students will work in groups of 2 or 3 to complete a variety of classroom and on-line application programs while developing the necessary interpersonal skills to work professionally with programmers, clients, and government agencies.

3. Information: the students will use information gained from lectures, readings, and lab activities to analyze real world problems and implement solutions using Python programming.

4. Systems: the students will understand their role as an software engineer working with a system analyst, to implement a software application, and maintain technical documents to ensure compliance with software engineering requirements.

5. Technology: the students will use on-line resources to search and learn about various case studies pertaining to design and implemention and to familiarize themselves with or review the various regulations that exist at a professional level in software engineering.

The course also addresses the SCANS skills and personal qualities:

1. Basic Skills: the students will use reading, writing, listening, and speaking skills to complete the course assignments. Emphasis will be placed upon proper technical writing skills which will include writing algorithms, flowcharts, Input-Process-Output (IPO) charts and applications, proper usage of the technical vocabulary words relevant to software engineering requirements.

2. Thinking Skills: the students will think creatively and critically to analyze a real world problems and synthesize reasonable and appropriate solutions, taking into account implementing a software system.

3. Personal Qualities: the students will follow the software engineering code of ethics in completing all their assignments and in all their interactions with their peers, professors, industry guest speakers, and other individuals with whom they will interact on a professional basis during their field trips or site visits. They will conduct themselves in a professional, responsible manner while exhibiting a strong work ethic and the highest standards of honesty and integrity.

XV. DISTANCE LEARNING COURSE OUTLINE ADDENDUM

CS M10P: Not Applicable

XVI.   GENERAL EDUCATION COURSE OUTLINE ADDENDUM

CS M10P: Not Applicable

XVII.   STUDENT MATERIALS FEE ADDENDUM

CS M10P: Not Applicable

XVIII.   REPEATABILITY JUSTIFICATION TITLE 5, SECTION 55041

CS M10P: Not Applicable

XIX.   CURRICULUM APPROVAL

Course Information:

Discipline: COMPUTER SCIENCE (CS)

Discipline Code and Number: CS M10P

Course Revision Category: New Course

Course Proposed By:

Originating Faculty: Esmaail Nikjeh 04/17/2017

Faculty Peer: _____

Curriculum Rep: Scarlet Relle 10/15/2017

Department Chair: _____

Division Dean: Mary Rees 04/17/2017

Approved By:

Curriculum Chair: Jerry Mansfield 11/10/2017

Executive Vice President: _____

Articulation Officer: Letrisha Mai 10/19/2017

Librarian: Mary LaBarge 10/31/2017

Implementation Term and Year: Fall 2018

Approval Dates:

Approved by Moorpark College Curriculum Committee: 11/07/2017

Approved by Board of Trustees (if applicable): 12/12/2017

Approved by State (if applicable): 02/16/2018