Syllabus (175 Credits) applicable to the students admitted to BE 3rd semester during the academic year 2020-21 and lateral entry students admitted during 2021-22

UMA391C	**Numerical Techniques and Integral Transforms**	**03-Credits**
Hrs/Week : 03		CIE Marks:50
Total Hours:40		SEE Marks:50

UNIT - I	**10 Hrs**
Numerical analysis - I: Introduction to root finding problems, Bisection Method, Newton-Raphson method. Finite differences, forward and backward difference operators (no derivations on relations between operators) Newton-Gregory forward and backward interpolation formulae. (Without proof), Lagrange's and Newton's divided difference interpolation formulae (without proof).	
UNIT – II	**10 Hrs**
Numerical analysis - II: Numerical differentiation using Newton's forward and backward formulae problems. Trapezoidal rule, Simpson's one third rule, Simpson's three eighth rule and Weddle's rule (no derivation of any formulae) problems. Euler's and Modified Euler's method, Runge-Kutta 4th order method.	
UNIT - III	**10 Hrs**
Fourier series: Periodic functions, Conditions for Fourier series expansions, Fourier series expansion of continuous and functions having finite number of discontinuities, even and odd functions. Half-range series, practical harmonic analysis.	
UNIT - IV	**10 Hrs**
Fourier transforms and z-transforms: Infinite Fourier transforms and inverse Fourier transforms- simple properties, Fourier sine and Fourier cosine transforms, Inverse Fourier sine and cosine transforms. Z-transforms-definition, standard forms, linearity property, damping rule, shifting rule-problems.	
Text Books: 1. Steven C. Chapra & Raymond P Canale, Numerical Methods for Engineers. 2. Dr. B. S. Grewal, Higher Engineering Mathematics, Khanna Publishers, New Delhi. 3. H. K. Das, Advanced Engineering Mathematics, S. Chand & company Ltd. Ram Nagar, New Delhi. 4. E Kreyszig, Advanced Engineering Mathematics, John Wiley & Sons.	

Code:UAI302C	**Data Structures and Applications**	**04 - Credits**
Hrs/Week : 04		CIE Marks:50
Total Hrs: 52		SEE Marks:50

Prerequisite: C programming

Course outcomes:

On the successful completion of this course, students are able to:

1. Summarize linear and nonlinear data structures concepts, searching and sorting techniques.
2. Analyze and implement different data structures, searching and sorting techniques.
3. Compare and contrast different types of data structures and searching and sorting methods.
4. Develop solutions for the given problem by using relevant data structures.

UNIT - I	**13 Hrs**
Introduction to Data Structures: Basic Concepts: Abstract data type: Atomic and composite data, Data type, Data structure, Abstract data type, Model for an abstract data type: ADT operations, ADT data structures, Pointer to void. **Pointer to Function**: Defining pointers to functions, Using pointers to functions. **Stacks**: **Basic stack operations**: Push, Pop, Stack top. **Stack linked list**: Implementation, Data structure, Stack head, Stack data node, Stack algorithms, Create Stack, Push Stack, Stack top, Empty Stack, Full Stack, Stack count, Destroy Stack, **C language implementation**: Insert data, Push Stack, Print Stack, Pop character. **Stack ADT**: Data structure, ADT implementations, Stack structure, Create stack, Push stack, Pop stack, Stack top, Empty stack, Stack count, Destroy stack. **Stack applications**: Reversing data, Reverse a list, Convert decimal to binary, Infix to postfix transformation, Evaluating postfix expressions, Stack implementation using array.	
UNIT – II	**13 Hrs**
Queues: **Queue Operations**: Enqueue, Dequeue, Queue front, Queue rear, Queue example. **Queue Linked list design**: Data structure, Queue head, Queue data node, Queue algorithms, Create queue, Enqueue, Dequeue, Retrieving queue data, Empty queue, Full queue, Queue count, Destroy queue. **Queue ADT**: Queue structure, Queue ADT algorithms, Queue Implementation using array, Queue Applications. **Sorting:** Selection, Insertion, exchange and quick sorts. **Searching**: Sequential, binary search, hashed list searches.	
UNIT - III	**13 Hrs**
General Linear lists: Basic operations, Insertion, Deletion, Retrieval, Traversal. **Implementation:** Data structure, Head node, Data node, Algorithms, Create list, Insert node, Delete node, List search, Retrieve node, Empty list, Full list, List count, Traverse list, Destroy list. **List ADT**: ADT functions, Create list, Add node, Internal insertion function, Remove node, Internal delete function, Search list, Internal search function, Retrieve node, Empty list, Full list, List count, Traverse, Destroy list. **Circular linked lists and Doubly linked lists**: Create list, add node, delete node, retrieve node, search list.	

UNIT - IV	**13 Hrs**
Non-Linear lists: Trees: Basic tree concepts: Terminology, User representation, **Binary trees**: Properties, Height of binary trees, Balance, Complete and Nearly complete binary trees, **Binary tree traversals:** Depth-first traversals, Breadth-first traversals, **Expression Trees**: Infix traversal, Postfix traversal, Prefix traversal, **Huffman code, General trees, Binary search trees**: Basic concepts, **BST operations**: Traversals, Searches, Insertion Find the smallest and largest node, BST search, Insertion, Deletion, **Binary search tree ADT**, Data structure, Head and node structure, Algorithms, Create a BST, Insert a BST, Internal insert function, Delete a BST, Internal delete function, Retrieve a BST, Internal retrieve function, Traverse a BST, Internal traverse function, Empty a BST, Full BST, BST count, Destroy a BST, Internal destroy function. **Graphs: Basic concepts**, **Operations**: Insert vertex, Delete vertex, Add edge, Delete edge, Find vertex, **Graph storage structures**: Adjacency matrix, Adjacency list.	

Text Book:

1) Behrouz A. Forouzan and Richard F. Gilberg, 2nd Edition, Cengage Learning Publisher, 2005. **Data Structure A Pseudocode Approach with C**, (Chapter 1(1.2,1.3,1.5), 2,3,4 (4.1-4.4), 5, 6(6.1-6.3), 7(7.1-7.3), 11(11.1-11.3),12(12.2-12.4) 13(13.1-13.3) Appendix F.

Reference Books:

1) Data Structures Using C, Aaron M. Tenanbaum, Yedidyah Langsam, Moshe J Augenstein Pearson Education.
2) Data Structures and Program Design in C, Robert Kruse, Bruce Leung, C. L. Tondo, Shashi Mogalla, 2nd Edition, Pearson Education.
3) Data Structures with C, Seymour Lipschutz, Schaum's outlines, MGH Education.
4) Data Structures Through C,Yeshwant Kanetkar , BPB publications.

Code:UAI303C	**Embedded Systems**	**04 - Credits**
Hrs/Week: 04		CIE Marks:50
Total Hrs: 52		SEE Marks:50

Course outcomes:
On the successful completion of this course, students are able to:

1. Comprehend the difference between microprocessor and microcontroller architectures.
2. Simulate, analyze and develop basic programs using assembly and C language.
3. Demonstrate the use of Timers, Counters, Interrupts through programs.
4. Demonstrate the use of serial ports through programs for developing basic communication systems.
5. Analyze a problem and formulate appropriate computing solution for microcontroller based embedded applications.

UNIT - I	**13 Hrs**
Boolean Algebra: Definition of Boolean algebra, Boolean algebra theorems, A two-valued Boolean algebra, Boolean formulas and functions, Canonical Formulas, Manipulations of Boolean formulas. Gates and Combinational networks: Incomplete Boolean functions and Don't care conditions, Additional Boolean operations and Gates. Simplification of Boolean Expressions: K-maps and The Quine-McCluskey method. **Logic Design with MSI Components, Flip- Flops, Counters:** Binary adders and subtractors, Decimal adders, Comparators, Decoders, Multiplexers. The basic Bi-stable element, Latches, Master-Slave flip-flops (Pulse-Triggered flip-flops), Edge triggered flip-flops, Characteristic equations, Registers, Counters, Design of synchronous counters.	
UNIT – II	**13 Hrs**
The 8051 Microcontrollers, Assembly Language Programming: Microcontrollers and Embedded systems, Overview of the 8051 family, Inside the 8051, Introduction to 8051 Assembly programming, Assembling and running an 8051 program, the program counter and ROM space in the 8051, 8051 data types and directives, 8051 flag bits and PSW register, 8051 register banks and stack, pin description of the 8051. **Jump, Loop and Call Instructions, I/O Port Programming**: Loop and Jump instructions, Call instructions, Time delay for various 8051 chips, 8051 I/O programming, I/O bit manipulation programming.	
UNIT - III	**13 Hrs**
8051 Addressing Modes, Arithmetic, Logic Instructions and Programs: Immediate and register addressing modes, Accessing memory using various addressing modes, Bit addresses for I/O and RAM, Extra 128-byte-on-chip RAM in 8052. Arithmetic instructions, Signed number concepts and arithmetic operations, Logic and compare instructions, Rotate instruction and data serialization, BCD, ASCII, and other application programs.	

UNIT - IV	**13 Hrs**
8051 Programming in C, Pin description of 8051: Data types and time delay in 8051 C, I/O programming in 8051 C, Logic operations in 8051 C, Data conversion programs in 8051 C, Accessing code ROM space in 8051 C, Data serialization using 8051 C. **8051 Timer Programming in Assembly and C**: Programming 8051 timers, counter programming, Programming timer 0 and 1 in 8051 C. **Interrupts Programming in Assembly and C**: 8051 interrupts, Programming timer interrupts, Programming external hardware interrupts, Programming the serial communication interrupt, Interrupt priority in the 8051/52, Interrupt programming in C. **MOTOR Control:** DC and Stepper Motors.	

Text Books:

1) Donald D. Givone, Digital Principles and Design, McGraw Hill Edition 2002.
2) Muhammed Ali Mazidi, Janice Gillispie Mazidi and Rolin D. Mckinlay, The 8051 Microcontroller and Embedded Systems Using Assembly and C, Pearson 2nd Edition, 2011.

Reference Books:

1) Leach and Malvino, Digital Principles and Applications, TMH, New Delhi, 2002.
2) Yarbrough J. M, Digital logic- Applications and Design, Thomson Learning, New Delhi, 2001.
3) Kenneth J. Ayala, The 8051 Microcontroller Architecture, Programming and Applications, 2nd Edition, Penram International, 1996.
4) Uma Rao and Andhe Pallavi, The 8051 Microcontroller Architecture, Programming and Applications, Pearson Education Sanguine.
5) V. Udayshankar, M. S. Mallikarjunaswamy, 8051 Microcontroller: Hardware, Software and Applications, McGrawHill, New Delhi.

Code: UAI304C	**Computer Organization**	**04 - Credits**
Hrs/Week:04		CIE Marks: 50
Total Hrs:52		SEE Marks:50

Course outcomes:

On the successful completion of this course, students are able to:

1. Describe the fundamental organization of a digital computer.
2. Explain the functional units and components of a computer.
3. Explain various addressing modes, instruction formats and program control statements and write assembly-level programs using simple machine instructions.
4. Distinguish the organization of various parts of a system memory hierarchy.
5. Describe fundamental concepts of pipelining and parallel processing.

UNIT - I	**10 Hrs**
Basic structure of computers: Computer types, Functional Units, Basic operational concepts, Bus structures. **Machine instructions and programs:** Numbers, Arithmetic operations and characters, Memory locations and addresses, Memory operations, Instructions and instruction sequencing, Addressing modes, Assembly language, assembler directives, number notation, Stacks and Queues, Subroutines, Encoding of machine instructions.	
UNIT – II	**10 Hrs**
Input/output organization: Accessing I/O devices, Interrupts-Interrupt hardware, Enabling and Disabling Interrupts, Handling multiple devices, controlling device requests, Exceptions, Direct memory access-bus Arbitrations, Buses-Asynchronous bus and Synchronous bus, Interface Circuits-Parallel port and serial port, Standard I/O Interfaces-Peripheral component interconnect Bus, SCSI bus, USB.	
UNIT - III	**10 Hrs**
The memory system: Some Basic concepts, Semiconductor RAM memories, Read only memories, speed, size, and cost, cache memories. **Arithmetic Unit:** Addition and subtraction of signed numbers, Design of fast adders, Multiplication of positive numbers, Signed operand multiplication, Fast multiplication. Integer Division, Floating point numbers and operations-IEEE standard for Floating point numbers, Arithmetic operations on Floating point numbers. Implementing Floating point operations.	
UNIT - IV	**10 Hrs**
Basic Processing Unit: Some fundamental concepts, Execution of complete instruction, Hardwired Control, Micro programmed control, Micro instructions. **Pipelining:** basic concepts, role of cache memory, pipeline performance. **Large computer systems**: forms of parallel processing, array processor, the structure of general purpose and multiprocessors. **Performance:** Processor Clock, Basic performance equation, pipelining and superscalar operations, Clock rate, Instruction set, compiler, performance measurement.	
Text Books: 1) Hamacher, Zvonko Vranesic, Safwat Zaky, 2002, Computer Organization, 5th Edition, MGH. (1.1-1.4, 2.1-2.5, 2.6.1, 2.6.3, 2.8-2.9, 2.12, 4.1, 4.2, 4.2.1-4.2.5, 4.4, 4.4.1, 4.5, 4.5.1-4.5.2, 4.6, 4.7, 5.1-5.5, 5.5.1, 6.1-6.7, 7.1-7.5, 7.5.1, 8.1, 8.1.1, 8.1.2, 12.1-12.3, 1.6)	
Reference Book: 1) J.P. Hayes, 1998, Computer Architecture and Organization, 3rd Edition, MGH. 2) William Stallings, 2007, Computer Organization and Architecture, 7th Edition, PHI.	

Code:UAI305C	**AI and its Applications**	**03-Credits**
Hrs/Week:03		CIE Marks:50
Total Hrs: 40		SEE Marks:50

Course outcomes:
On the successful completion of this course, students are able to:

1. Demonstrate fundamental understanding of the history of artificial intelligence (AI) and its foundations.
2. Demonstrate proficiency in usage of hardware and software platforms for AI based applications.
3. Demonstrate awareness and understanding of various applications of AI techniques.
4. Demonstrate an ability to share in discussions of AI, its current scope and limitations, and societal implications.

UNIT - I	**10 Hrs**

Introducing AI: Defining the Term AI, Discerning intelligence, Discovering four ways to define AI, Understanding the history of AI, Starting with symbolic logic at Dartmouth, Continuing with expert systems, Overcoming the AI winters, Considering AI uses, Avoiding AI Hype, Connecting AI to the underlying computer.
Defining the role of data: Finding data ubiquitous in this age, Understanding Moore's implications, Using data everywhere, Putting algorithms into action.
Considering the use of algorithms: Understanding the role of algorithms, Understanding what algorithm means, starting from planning and branching, Playing adversarial games, Using local search and heuristics, Discovering the learning machine, Leveraging expert systems, Introducing machine learning, Touching new heights.
Pioneering specialized hardware: Relying on standard hardware, Understanding the standard hardware, Describing standard hardware deficiencies, Using GPUs, Considering the Von Neumann bottleneck, Defining the GPU, Considering why GPUs work well, Creating a specialized processing environment, Increasing hardware capabilities, Adding specialized sensors, Devising methods to interact with the environment.

UNIT – II	**10 Hrs**

Seeing AI uses in computer applications: Introducing common application types, Using AI in typical applications, Realizing AI's wide range of fields, Considering the Chinese Room argument, Seeing how AI makes applications friendlier, Performing corrections automatically, Considering the kinds of corrections, Seeing the benefits of automatic corrections, Understanding why automated corrections don't work, Making suggestions, Getting suggestions based on past actions, Getting suggestions based on groups, Obtaining the wrong suggestions, Considering AI-based errors.
Using AI to address medical needs: Implementing portable patient monitoring, Wearing helpful monitors, Relying on critical wearable monitors, Using movable monitors, Making humans more capable, Using games for therapy, Considering the use of exoskeletons, Addressing special needs, Considering the software-based solutions, Relying on hardware augmentation, Seeing AI in prosthetics, Completing analysis in new ways, Devising new surgical techniques, Making surgical suggestions, Assisting a surgeon, Replacing the surgeon with monitoring, Performing tasks using automation, Working with medical records,

Predicting the future, Making procedures safer, Creating better medications, Combining robots and medical professionals.

Relying on AI to improve human interaction: Developing new ways to communicate, Creating new alphabets, Automating language translation, Incorporating body language, Exchanging ideas, Creating connections, Augmenting communication, Defining trends, Using multimedia, Embellishing human sensory perception, Shifting data spectrum, Augmenting human senses.

UNIT - III	**10 Hrs**

Performing data analysis for AI: Defining data analysis, Understanding why analysis is important, Reconsidering the value of data, Defining machine learning, Understanding how machine learning works. Understanding the benefits of machine learning, Being useful; being mundane, Specifying the limits of machine learning, Considering how to learn from data, supervised learning, Unsupervised learning, Reinforcement learning.

Employing machine learning in AI: Taking many different roads to learning, Discovering five main approaches to AI learning, Delving into the three most promising AI learning, approaches, Awaiting the next breakthrough, Exploring the truth in probabilities, Determining what probabilities can do, Considering prior knowledge, Envisioning the world as a graph, Growing trees that can classify, Predicting outcomes by splitting data, Making decisions based on trees, Pruning overgrown trees.

Developing robots and flying with drones: Defining robot roles, Overcoming the sci-fi view of robots, Knowing why it's hard to be a humanoid, Working with robots, Assembling a basic robot, Considering the components, Sensing the world, Controlling a robot, Acknowledging the state of the art, Flying unmanned to missions, Meeting the quad-copter, Defining uses for drones, Seeing drones in non-military roles, Powering up drones using AI, Understanding regulatory issues.

UNIT - IV	**10 Hrs**

Understanding the Non starter Application: Using AI where it won't work, Defining the limits of AI, Applying AI incorrectly, Entering a world of unrealistic expectations, Considering the effects of AI winters, Understanding the AI winter, Defining the causes of the AI winter, Rebuilding expectations with new goals, Creating solutions in search of a problem, Defining a gizmo, Avoiding the infomercial, Understanding when humans do it better, Looking for the simple solution.

Seeing AI in space: Observing the universe, Seeing clearly for the first time, Finding new places to go, Considering the evolution of the universe, Creating new scientific principles, Performing space mining, Harvesting water, Obtaining rare earths and other metals, Finding new elements, Enhancing communication, Exploring new places, Starting with the probe, Relying on robotic missions, Adding the human element, Building structures in space, Taking your first space vacation, Performing scientific investigation, Industrializing space, Using space for storage.

Adding new human occupations: Living and working in space, Creating cities in hostile environments, Building cities in the ocean, Creating space-based habitats, Constructing moon-based resources, Making humans more efficient, Fixing problems on a planetary scale, Contemplating how the world works, Locating potential sources of problems, Defining potential solutions, Seeing the effects of the solutions, Trying again.

Text Books:

1) John Paul Mueller and Luca Massaron, Artificial Intelligence for Dummies, John Wiley and Sons, 2018.

Reference Books:

1) Utpal Chakraborthy, Artificial Intelligence for all, BPB Publications, Feb. 2020.
2) Praphat Kumar, Artificial Intelligence, BPB Publications, Jan. 2019.
3) Nils J. Nilsson, The Quest for Artificial Intelligence: A History of Idea and Achievements, Stanford University, Camridge University Press, 2010.
4) Bernard Marr, Artificial Intelligence: How 50 Successful Companies used Artificial Intelligence to solve Problems, Wiley Publications, 2019.

Code: UAI306L	**Problem Solving with Python Lab.**	**02-Credits**
Hrs/Week: (02+02)		CIE Marks:50
Total Hours:28		SEE Marks:50

Course outcomes:
On the successful completion of this course, students are able to:

1. Write, test, and debug simple Python programs with conditionals and loops.
2. Develop Python programs step-wise by defining functions and calling them.
3. Use Python lists, Tuples, Dictionaries for representing compound data.
4. Read and Write data from/to files in Python.
5. Display drawing in the canvas and design GUI using widgets of python.

UNIT - I	**07 Hrs**

Data types in python: Comments in python, Doc strings, How python sees variables, Data types in python, built in data type, bool data type, Sequences in python, Sets, Literals in python, Determining the data type of a variable, user defined data types , constants in data type, Identifiers and reserved words, Naming conventions in python. **Operators in Python:** Operator, arithmetic operator, using python interpreter as a calculator, assignment operator, unary minus operator, relational operators, logical operators, boolean operators, bitwise operators, membership operators, identity operators, operator precedence and Associativity, Mathematical functions, using IDLE window, using command line window, executing at system prompt. **Input and Output:** Output statements, Input statements, Command Line arguments. **Control Statements:** Control statements, the if statement, a word on indentation, the if else statement, the if elif else statement, the while loop, the for loop, infinite loops, nested loops, the else suite, the break statement, the continue statement, the pass statement, the assert statement, the return statement.

UNIT – II	**07 Hrs**

Arrays in Python: Array, advantages of array, creating an array, importing the array module, indexing and slicing on array, processing the arrays, types of arrays, working with arrays using numpy, creating arrays using array, creating arrays using linespace, creating arrays using log space, creating arrays using arrange function, creating arrays using zeros and ones functions, mathematical operations on arrays, comparing arrays, aliasing the arrays, viewing and copying arrays, slicing and indexing in numpy arrays, dimensions of arrays, attributes of an arrays, the reshape method, the flatten method, working with multi-dimensional arrays, indexing in multi dimensional arrays, slicing the multi dimensional arrays, matrices in numpy, getting diagonal elements of a matrix, finding maximum and minimum elements, finding sum and average of elements, products of elements, sorting the matrix, transpose of a matrix, matrix addition and multiplication, random numbers. **Strings and Characters:** Creating strings, length of a string, indexing in strings, slicing the strings, repeating the string, concatenation of strings, checking membership, comparing strings, removing spaces from a string, finding sub string, counting substrings in a string, strings are immutable, replacing a string with another string, splitting and joining stings, changing case of a string, checking starting and ending of a string, string testing methods, formatting the strings, working with characters, sorting strings, searching in the strings, finding number of characters and words, inserting sub string into a string. **Functions:** Difference between a function and a method, defining a function, calling a function, Returning Results from a

function, Returning multiple values from a function, functions are first class objects, pass by object reference, Formal and actual arguments, positional arguments, keyword arguments, default arguments, variable length arguments, local and global variables, the global keyword, passing a group of elements to a function, recursive functions, anonymous functions or lambdas, F unction decorators, generators, structured programming, creating our own modules in python, the special variable name.

UNIT - III	**07 Hrs**

Lists and Tuples: Lists, creating lists using range function, updating the elements of a list, concatenation of two lists, repetition of lists, membership in lists, aliasing and cloning lists, methods to process lists, finding biggest and smallest elements in a list, sorting the list elements, number of occurrences of an elements in the list, finding common elements in two lists, storing different types of data in a list, nested list, nested lists as matrices, list comprehensions, Tuples, creating tuples, accessing the tuple elements, basic operations on tuples, functions to process tuples, nested tuples, inserting elements in a tuple, modifying elements of a tuple, deleting elements from a tuple. **Dictionaries:** Operations on dictionaries, dictionary methods, using of loop with dictionaries, sorting the elements of a dictionary using lambdas, converting lists into dictionary, converting strings into dictionary, passing dictionaries to functions, ordered dictionaries. **Regular Expressions in python:** Regular expressions, sequence characters in regular expressions, quantifiers in regular expressions, special characters in regular expressions, using regular expressions on files, retrieving information from a HTML file. **Exceptions:** Errors in a python program, exceptions, exception handling, types of exceptions, the except block, the assert statement, user-defined exceptions, logging the exceptions.

UNIT - IV	**07 Hrs**

Files in Python: Files, types of files in python, opening a file, closing a file, working with text files containing strings, knowing whether a file exists or not, working with binary files, the with statement, pickle in python, the seek and tell methods, Random accessing of binary files, random accessing of binary files using mmap, zipping and unzipping files, working with directories, running other programs from python program. **Date and Time:** The epoch, date and time now, combining date and time, formatting dates and times, finding durations using time delta, comparing two dates, sorting dates, stopping execution temporarily, knowing the time taken by a program, working with calendar module. **Graphical user Interfaces:** GUI in python, the root window, fonts and colors, working with containers, canvas, frame, widgets, button widget, arranging widgets in the frames, label widget, message widget, text widget, scrollbar widget, check button widget, radio button widget, entry widget, spin box widget, list box widget, menu widget.

Textbooks:

1. Dr. R. Nageswawa Rao, Core Python Programming, Dreamtech press, 2nd Edition 2018 (Chapter Numbers: 3,4,5,6,7, 8,9,10,11,16,17,18,22).

Reference Books:

1. Gowrishankar S. Veena A, Introduction to Python Programming, CRC Press Taylor & Francis Group, 1st Edition 2019.
2. Michael Urban and Joel Murach, Mike Murach Elizabeth Drake, Python Programming, 1st Edition, 2016.