CS620/DASC600 – Introduction to Data Science and Analytics
Fall 2023

COURSE INFORMATION

Instructors : Dr. Sampath Jayarathna, http://www.cs.odu.edu/~sampath/
Office : ECSB 3109, sampath@cs.odu.edu , Phone: (757) 683-7787
Recitation : Thursday, 5:45 PM – 7:00 PM (Online Zoom link)
Office Hours : Tuesday, 12PM – 1PM, In-person/Zoom or email me for an appointment

Teaching Assistant:
Yasith Jayawardana, yasith@cs.odu.edu , Office: ECSB 2100A

Content : https://canvas.odu.edu/
All sections of this course content will be delivered fully online, asynchronously via Canvas weekly modules.

Piazza : https://piazza.com/odu/Fall2023/cs620/home

Prerequisites : There are no specific course prerequisites for this course. But, I expect you to be comfortable coding in Python and its associated libraries, and knowledge in linear algebra and statistics.

WHAT IS THIS COURSE ABOUT?

Data science is an interdisciplinary blend of the analytical, computational, and statistical skills necessary to extract knowledge from large and complex sets of data. The proliferation of such data has led to an acute shortage of students with data science skills in the local, national, and global economy.

This course will introduce students to this rapidly growing field of Data Science and equip them with some of its basic principles and tools as well as its general mindset. Students will learn concepts, techniques, and tools they need to deal with various facets of data science practices. Cross-listed with DASC 600.

WHAT WILL YOU GET FROM THIS COURSE?

- Define and explain the key concepts and models relevant to data science.
- Understand the processes of data science: identifying the problem to be solved, data collection, preparation, modeling, evaluation and visualization.
- Develop an appreciation of the many techniques for data modeling
- Be comfortable using commercial and open-source tool such as python and associated libraries for data analytics and visualization.

REQUIRED/OPTIONAL MATERIALS:

- **Required textbook**. No textbook is required. All the key course content will be documented in Canvas Modules.

- **List of optional but recommended materials.** You may find some of these optional textbooks helpful, though none are required:
  - A Hands-On Introduction to Data Science by Chirag Shah, Cambridge University Press, April 2, 2020
  - Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, By William McKinney, O'Reilly; 2 edition (October 20, 2017)
  - Data Science from Scratch: First Principles with Python By Joel Grus, O'Reilly 1st edition, 2015

COURSE STRUCTURE AND TENTATIVE COURSE SCHEDULE

Structure: Course is divided into two units with each unit having several modules. Unit-I is the programming unit that covers python, and its associated libraries required for data science activities. Unit-II is the core unit that covers data science concepts and techniques.

This is a hands-on course with a number of programming activities, assignments, and a final data project. Participation in discussion forum (piazza) is mandatory and will also be graded. See discussion participation section for more details. The course project is a major component of this course, which will enable students to apply their knowledge acquired in the course to develop and implement a data science application on one or more of the technologies covered in Unit-I and II.

Duration: Unit-1 is covered in 7 weeks, Unit-II is covered in 7 weeks. The course project overlaps with Unit-I and II and is spread over the semester. See the topics below for more information.

Topics: The tentative topics are as follows.

Unit-I:

Week 1: Syllabus and Introductions
Week 2: Python
Week 3: NumPy
Week 4: Pandas
Week 5: Data Wrangling
Week 6: Semi-Structured Data
Week 7: NoSQL

Unit-II:

Week 8: Text Data Analysis and Inference
Week 9: Pattern Mining
Week 10: Machine Learning on Data
Week 11: Machine Learning on Data
Week 12: Weka
Week 13: Evaluations
Week 14: Delivering Results

Week 15: Final Exam (December 06-07, via Canvas)

Module Activity:

- Watch module overview video from the instructor
- Read instruction material
- Complete weekly course activities and submit all required work to piazza/Canvas.
- Complete homework assignments (Note that not all modules have homework assignments)
- Participate in piazza discussion forum. This is mandatory and will be graded, see discussion participation section for details

WHAT YOU CAN EXPECT FROM ME:

I am committed to supporting students with disabilities. If you have challenges related to these issues or others, I want to work with you to help you succeed. Please contact me, since only you can properly communicate your situation to me.

WHAT YOU CAN GIVE TO THE CLASS:

It is extremely important for you to be engaged in the course; otherwise, you will wonder what happened to your tuition dollars. So, I encourage you to setup online meetings, ask questions and actively participate at the piazza discussion forum.

COMMUNICATION

Piazza: All questions will be fielded through Piazza. The primary benefit is that for many questions everyone can see the answer and other students can answer as well. I will endorse good student responses. Additionally, I expect you to actively participate in online discussions at Piazza. You can post public or private messages that can only be seen by the instructor. You will be signed up with your odu email, but you may switch to another email.

Canvas: Cavans will be used primarily for course weekly module content and grade dissemination.

Email: *Again, email should only be used in rare instances*, I will probably point you back to Piazza if you have a question related to course materials and/or relevant to other students in the class.

If you send email to me (for any urgent matter such as health issue etc.,), please be sure to include your name and the course number in the body of the e-mail. You should also use an appropriate subject line that looks like “CS620-HW2” etc. Failure to follow these guidelines may result in delayed response.

COURSE ACTIVITIES

The scores you receive on the various graded tasks in the class will be weighted as follows:

40%	Data project
20%	Final Exam
25%	Homework Assignments (5)
15%	in-class activities + discussion forum interactions
100%	Your Total Score for the class

Final Exam: Final examination will be a comprehensive (covering all the modules), online (delivered via Canvas) exam and will be scheduled during the last week of the class. On the week before final exam, I will post a study guide that will help students to prepare for the written examination. You may have one standard 8.5" by 11" piece of paper with any notes you deem appropriate or significant (front and back) for the final exam.

Homework: We will have several homework assignments, worth 25% of your overall grade.

Data Project: The data project is an opportunity to tackle a more challenging data science activity. Details, requirements, and submission information will be on the project section of the course web page. For the project, you will work **individually or team of 2-3 students** on a problem of your choosing that is interesting, significant, and relevant to data science. The ultimate goal of your course project is to tackle some

interesting real-world problem. All members of a group will receive the same grade on group work. Therefore, it is in your interest to choose other group member (ideally, first week of the class) who have the same goal in the class as you do. It is also in your interest to work together and ensure that all tasks are completed effectively. Your scores on group work may be adjusted based on your contribution. The goal of your data project is to apply the techniques learn in each week of the class towards your dataset (exploration, wrangling, machine learning, visualization). We are going to use Google Colab (Colaboratory) (https://colab.research.google.com/), a free Jupyter notebook environment that requires no setup and runs entirely in the cloud. With Colaboratory you can write and execute code, save and share your analyses, and access powerful computing resources, all for free from your browser.

Class Activities/Discussion Forum interactions: Class activities and participation in the discussion are both important to your success in the course. As one measure of your participation and course preparation, we will have class activities related to lecture topics to supplement the learning.

GRADES

Final course grades are based on the overall average. Overall class grade (not the individual grade) windows may be increased in size if the instructor finds it appropriate. Final score in % will be rounded to the nearest whole number. Assigning + or – grades may be made at instructor's discretion.)

A: 94-100, A-:90-93, B+:87-89, B: 84-86, B-:80-83, C+:78-79, C: 74-77, C-:70-73, Fail (Grade F): 0-69

Grading correction: The assignment or exam grading correction requests should be sent to the instructor within 1 week of receiving the grade, or before the end of the semester, whichever comes first. After that, your grade will not be adjusted. If you find a mistake in grading, please let the instructor know. Your grade will not be lowered.

There is no separate grading scale for PhD students, but PhD students will typically be held to a higher standard.

COURSE POLICIES:

Attendance: Since this is an on-line course, there is no mandatory attendance policy. However, students are expected to actively participate in the piazza discussions, class activities, homework submissions, and Google Colab project writing. Each of these components is graded and counted towards the final grade.

Recitation Attendance: Students are expected to attend at the scheduled time.

Online Classroom Conduct (Netiquette): Students are expected to follow good Netiquette rules. Netiquette is the accepted behavior for online participation. The following is a list of general guidelines for this course:

- Check your grammar and spelling
- Keep your comments focused on the topic
- Strive to write succinctly and clearly
- Share your knowledge and include supportive evidence for your comments
- Do not use all capital letters as that is viewed as shouting
- Avoid flaming—disrespectful language is unacceptable

Select the link to find more information on Netiquette.

Tests, Make-ups, and Late Policies: Due dates will be set to give ample time for completion of the assignments and will not be extended save for the unexpected and unlikely major, long-lived catastrophe. Start projects and homework early--last minute computer malfunctions will not be accepted as a reason for delaying a due date. Changes to a submission's due dates will be avoided because they are unfair to those students who have organized their time to complete the assigned work. Individual accommodations will be discussed if you have a valid medical excuse.

Unless otherwise specified by the instructor, Final Exam will be comprehensive, covering material from the entire course. There are no makeups or rescheduling of exam unless you have a plausible reason with appropriate document or verification. Rescheduling of exams must be arranged at least 3 days in advance. An exam missed without an acceptable excuse will be recorded as a grade of zero (0).

For Homework assignments, each late submission will incur a 5 points penalty per day. A missed submission without an acceptable excuse will be recorded as a grade of zero (0). No submission will be accepted after 3rd day and will be recorded as a grade of zero (0).

There will be no makeup for homework assignments or class activities.

ACADEMIC OFFENSES

Class Conduct

The following standards are intended to define acceptable behavior that preserves academic integrity and ensures that students have optimum environmental conditions for effective learning.

- Students should notify instructors in advance when falling behind. In the event of an emergency that might affect the progress in the course, instructors must be notified as soon as possible.
- Students will activate their Old Dominion email accounts and check them before each class. If the student chooses to have his/her messages forwarded to another account, it is the student's responsibility to take the necessary steps to have them forwarded.
- Offensive language, gestures and the like are disrespectful and disruptive to the teaching-learning process.

Academic Integrity

Old Dominion University is committed to students' personal and academic success. In order to achieve this vision, students, faculty, and staff work together to create an environment that provides the best opportunity for academic inquiry and learning. All students must be honest and forthright in their academic studies. Your work in this course and classroom behavior must align with the expectations outlined in the Code of Student Conduct, which can be found at www.odu.edu/oscai. The following behaviors along with classroom disruptions violate this policy, corrupt the educational process, and will not be tolerated.

Cheating: Using unauthorized assistance, materials, study aids, or other information in any academic exercise.

Plagiarism: Using someone else's language, ideas, or other original material without acknowledging its source in any academic exercise.

Fabrication: Inventing, altering or falsifying any data, citation or information in any academic exercise.

Facilitation: Helping another student commit, or attempt to commit, any Academic Integrity violation, or failure to report suspected Academic Integrity violations to a faculty member.

Academic dishonesty will be reported to the Office of Student Conduct & Academic Integrity and may result in sanctions up to and including expulsion from the University.

Plagiarism

No plagiarism will be tolerated under any circumstances. As faculty, I am bound to report any instances of plagiarism. All cases are heard before the honor council. If found guilty, the student automatically receives a failing grade (F) in the course, and a notice is entered into the permanent record for a period of time.

College Class Conduct

The following standards are intended to define acceptable classroom behavior that preserves academic integrity and ensures that students have optimum environmental conditions for effective learning.

- Students must turn off cell phones during class or have them set to vibrate mode.
- Classes are expected to begin on time, and students will respect the time boundaries established by the professor. If classroom doors are locked, students may not knock or seek entrance in other ways.
- Students should notify instructors in advance when a class will be missed. In the event of an emergency that causes a class to be missed, instructors must be notified as soon as possible.
- Instructors may require that cell phones and other electronic devices be left on their desks during tests or examinations.
- Students must not engage in extraneous conversations during classes. Such acts are considered to be violations of the Code of Student Conduct.
- Students will activate their Old Dominion e-mail accounts and check them before each class. If the student chooses to have his/her messages forwarded to another account, it is the student's responsibility to take the necessary steps to have them forwarded.
- Consumption of food and drink during class is prohibited, except when the professor has specifically approved of such acts.
- Offensive language, gestures and the like are disrespectful and disruptive to the teaching-learning process.

Honor Code

The Old Dominion University Honor Code will be strictly enforced. By attending Old Dominion University, you have signed a pledge accepting the responsibility to abide by the following Honor Code found at Office of Student Conduct and Academic Integrity.

"We, the students of Old Dominion University, aspire to be honest and forthright in our academic endeavors. Therefore, we will practice honesty and integrity and be guided by the tenets of the Monarch Creed. We will meet the challenge to be beyond reproach in our actions and our words. We will conduct ourselves in a manner that commands the dignity and respect that we also give to others."

This is an institutional policy approved by the Board of Visitors. The University Honor Code applies to all assignments.

Honor Pledge

"I pledge to support the honor system of Old Dominion University. I will refrain from any form of academic dishonesty or deception, such as cheating or plagiarism. I am aware that as a member of the academic community, it is my responsibility to turn in all suspected violators of the honor system. I will report to Honor Council hearings if summoned."

By attending Old Dominion University you have accepted the responsibility to abide by this code. This is an institutional policy approved by the Board of Visitors. For more information, please visit Policies and Student Responsibilities.

EDUCATIONAL ACCESSIBILITY

In compliance with PL94-142 and more recent federal legislation affirming the rights of disabled individuals, provisions will be made for students with special needs on an individual basis. The student must be identified by the university and provide a letter from the Office of Educational Accessibility (OEA), located at 1021 Student Success Center. Any accommodations will be based upon written guidelines from the Office of Educational Accessibility (OEA). All students are expected to fulfill all course requirements.

Old Dominion University is committed to ensuring equal access to all qualified students with disabilities in accordance with the Americans with Disabilities Act. The Office of Educational Accessibility (OEA) is the campus office that works with students who have disabilities to provide and/or arrange reasonable accommodations.

If you experience a disability which will impact your ability to access any aspect of my class, please present me with an accommodation letter from OEA so that we can work together to ensure that appropriate accommodations are available to you.

If you feel that you will experience barriers to your ability to learn and/or testing in my class but do not have an accommodation letter, please consider scheduling an appointment with OEA to determine if academic accommodations are necessary.

The Office of Educational Accessibility is located at 1021 Student Success Center and their phone number is (757)683-4655. Additional information is available at the OEA Website.

OTHER POLICIES

University Email Policy: Electronic messaging systems and communication services are provided by Old Dominion University for the purpose of enhancing productivity and maintaining effective communication.

Old Dominion University employees, students, employees of affiliated organizations, and guests, volunteers, and researchers who are provided official email accounts must activate and maintain regular access to these accounts. These accounts must be used to send and receive electronic communications related to official University business.

Failure to access the email account will not exempt individuals from their responsibility of being aware of and meeting requirements and responsibilities included in electronic communications.

Message content is the sole responsibility of the individual sending the message and users must adhere to University Policy 3500 Use of Computing Resources, and Information Technology Standard 09.1.0, Acceptable Use Standard. Users are also encouraged to practice generally accepted online etiquette.

Instructors retain the discretion of establishing class expectations for email and other electronic messaging communication as a part of the course requirements.

Alternative messaging services should be arranged in cases where users' access to information technology resources is limited or unavailable.

Incomplete: Documented illnesses, deaths in family, car accidents, or other traumatic occurrences call for flexibility and good judgment on the part of the student and instructor. These situations are rare and are handled individually. Should such a situation occur, students MUST contact Student Outreach & Support.

Email oducares@odu.edu or by phone 757-683-3442 to acquire the necessary documentation. An incomplete grade will only be given under the following circumstances

- The student has completed ½ or more of the course requirements with a C or better
- There is legitimate deficiency due to the illness or emergencies deemed acceptable to the instructor
- There is not neglect on the student's part.

Withdrawal: A syllabus constitutes an agreement between the student and the course instructor about course requirements. Participation in this course indicates your acceptance of its teaching focus, requirements, and policies. Please review the syllabus and the course requirements as soon as possible. If you believe that the nature of this course does not meet your interests, needs or expectations, if you are not prepared for the amount of work involved - or if you anticipate that the class meetings, assignment deadlines or abiding by the course policies will constitute an unacceptable hardship for you - you should drop the class by the drop/add deadline, which is located in the ODU Schedule of Classes. For more information, please visit the Office of the Registrar.

STUDENT ACKNOWLEDGMENT

"I, ASHISH VERMA 01266453, have completely read this syllabus and understand and agree to the course requirements".

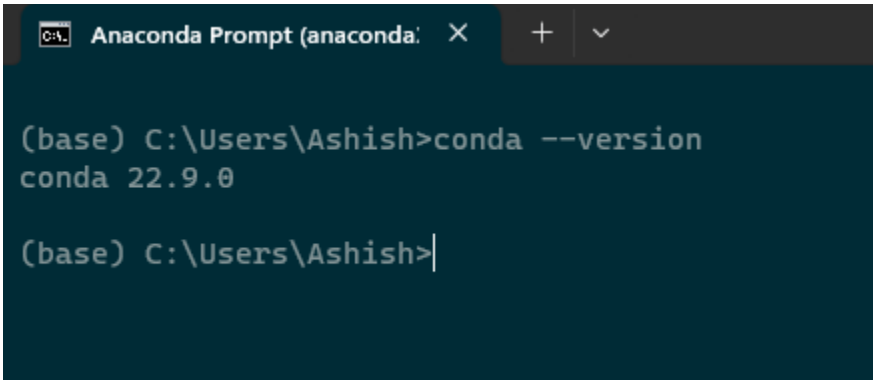
Ashish Verma 09/02/2023

Signature and Date

CS620
HW1
@Ashish_Verma_and_01266453

Module 2

Anaconda Installation on Windows Machine



__✓__ I certify that I have installed Anaconda or similar python environment and have practiced python by following the Module 2 as shown below

Variables and Objects

CS620
HW1
@Ashish_Verma_and_01266453

```
IPython  -- An enhanced Interactive Python.

In [1]: x = 5

In [2]: x
Out[2]: 5

In [3]: x = [1,3,5]

In [4]: x
Out[4]: [1, 3, 5]

In [5]: X ='python'

In [6]: X
Out[6]: 'python'

In [7]: y = x

In [8]: y
Out[8]: [1, 3, 5]

In [9]: a,b = 5,7

In [10]: a,b
Out[10]: (5, 7)

In [11]:
```

Input and Output

```
In [11]: print('This is the english sentence')
This is the english sentence

In [12]: person = input("Enter your age")
Enter your age 35

In [13]: print("Age is \n",person)
Age is
  35

In [14]:
```

CS620
HW1
@Ashish_Verma_and_01266453

Arithmetic and Complex Numbers

```
IPython  -- An enhanced Interactive Python.

In [1]: a = 5 + 2

In [2]: a
Out[2]: 7

In [3]: b = 9.0 - 3.0

In [4]: b
Out[4]: 6.0

In [5]: c = 5 * 2

In [6]: c
Out[6]: 10

In [7]: d = 5**2

In [8]: d
Out[8]: 25

In [9]: e = 5%2

In [10]: e
Out[10]: 1
```

CS620
HW1
@Ashish_Verma_and_01266453

```
In [11]: f = 7 /2

In [12]: f
Out[12]: 3.5

In [13]: f = 7//2

In [14]: f
Out[14]: 3

In [15]: f = 7/2.

In [16]: f
Out[16]: 3.5

In [17]: f = 7/float(3)

In [18]: f
Out[18]: 2.3333333333333335

In [19]: f = int(7/2)

In [20]: f
Out[20]: 3
```

f = 1+2.56j

print(f.real,f.imag)

1.0 2.56

CS620
HW1
@Ashish_Verma_and_01266453

Strings

```
In [24]: single_quoted_string
Out[24]: 'MSDSA'

In [25]: double_quoted_string = "MSDA"

In [26]: double_quoted_string
Out[26]: 'MSDA'

In [27]: escaped_string = 'Isn\'t is  python code'

In [28]: escaped_string
Out[28]: "Isn't is  python code"

In [29]: long_string = 'This is the example \
    ...: of long \
    ...: string'

In [30]: long_string
Out[30]: 'This is the example of long string'
```

```
In [32]: multi_line_string ="""This is the example of comments"""

In [33]: multi_line_string
Out[33]: '"This is the example of comments'

In [34]:
```

String Concatenation

CS620
HW1
@Ashish_Verma_and_01266453

```
In [34]: str1 ='Ashish'

In [35]: str2 ='Verma'

In [36]: str1 + str2
Out[36]: 'AshishVerma'

In [37]: s = 3 * 'un' + 'ium'

In [38]: s
Out[38]: 'unununium'
```

Collection Data Types

1. **List**: a collection which is ordered and *changeable*. Allows duplicate members.
2. **Tuple**: a collection which is ordered and *unchageable*. Allows duplicate members.
3. **Set**: a collection which is unordered and unindexed. No duplicate members.
4. **Dictionary**: a collection which is unordered, changeable and indexed. No duplicate members.

```
In [39]: x = [0,1,2,3,4,5,6,7]

In [40]: x
Out[40]: [0, 1, 2, 3, 4, 5, 6, 7]

In [41]: len(x)
Out[41]: 8

In [42]: sum(x)
Out[42]: 28

In [43]: x[1]
Out[43]: 1

In [44]: x[-1]
Out[44]: 7
```

CS620
HW1
@Ashish_Verma_and_01266453

```
In [46]:  one_to_four
Out[46]: [1, 2, 3, 4]

In [47]: first_three = x[:3]

In [48]: first_three
Out[48]: [0, 1, 2]

In [49]:  last_three = x[-3:]

In [50]: last_three
Out[50]: [5, 6, 7]

In [51]: three_to_end = x[3:]

In [52]: three_to_end
Out[52]: [3, 4, 5, 6, 7]

In [53]: no_first_last = x[1:-1]

In [54]: no_first_last
Out[54]: [1, 2, 3, 4, 5, 6]
```

CS620
HW1
@Ashish_Verma_and_01266453

```
In [55]: copy_of_x = x[:]

In [56]: copy_of_x
Out[56]: [0, 1, 2, 3, 4, 5, 6, 7]

In [57]: x = [1,2,3]

In [58]: y = [5,6,7]

In [59]: x.extends(y)
Traceback (most recent call last):

  Cell In[59], line 1
    x.extends(y)

AttributeError: 'list' object has no attribute 'extends'


In [60]: x.extend(y)

In [61]: x
Out[61]: [1, 2, 3, 5, 6, 7]

In [62]:
```

CS620
HW1
@Ashish_Verma_and_01266453

```
In [4]: x,y = [1,2]

In [5]: x
Out[5]: 1

In [6]: y
Out[6]: 2

In [7]: [x,y] = 1,2

In [8]: x
Out[8]: 1

In [9]: y
Out[9]: 2

In [10]: _,y = [3,4]

In [11]: y=4

In [12]:
```

CS620
HW1
@Ashish_Verma_and_01266453

Strings can also be sliced. But they cannot be modified (they are immutable).

```
In [12]: s = 'university'

In [13]: s[0]
Out[13]: 'u'

In [14]: s[:2]
Out[14]: 'un'

In [15]: s[-2"]
  Cell In[15], line 1
    s[-2"]
        ^
SyntaxError: unterminated string literal (detected at line 1)


In [16]: s[-2:]
Out[16]: 'ty'

In [17]: s[1:-1]
Out[17]: 'niversit'

In [18]:
```

Tuples--similar to lists, but immutable. Note: tuple is defined by comma, not (), which is only used for convenience. So a=(1) is not a tuple, but a = (1,) is.

CS620
HW1
@Ashish_Verma_and_01266453

```
In [20]: a_tuple
Out[20]: (0, 1, 2, 3, 4)

In [21]: Other_tuple
Out[21]: (3, 4)

In [22]: Another_tuple = tuple([0, 1, 2, 3, 4])

In [23]: Another_tuple
Out[23]: (0, 1, 2, 3, 4)

In [24]: Another_tuple[2]=6
Traceback (most recent call last):

  Cell In[24], line 1
    Another_tuple[2]=6

TypeError: 'tuple' object does not support item assignment


In [25]:
```

Dictionaries

CS620
HW1
@Ashish_Verma_and_01266453

```
In [25]: grades = {'Joel':80,'Tim':95}

In [26]: grades
Out[26]: {'Joel': 80, 'Tim': 95}

In [27]: joel_grades = grades['Joel']

In [28]: joel_grades
Out[28]: 80

In [29]: students = len(grades)

In [30]: students
Out[30]: 2

In [31]: kates_grades = grades.get('Tim',0)

In [32]: kates_grades
Out[32]: 95
```

Get all items

```
In [33]: all_keys = grades.keys()

In [34]: all_keys
Out[34]: dict_keys(['Joel', 'Tim'])

In [35]: all_values = grades.values()

In [36]: all_values
Out[36]: dict_values([80, 95])

In [37]: all_pairs = grades.items()

In [38]: all_pairs
Out[38]: dict_items([('Joel', 80), ('Tim', 95)])
```

CS620
HW1
@Ashish_Verma_and_01266453

Control Flow

```
IPython  -- An enhanced Interactive Python.

In [1]: a  = [1,2,3,4,5]

In [2]: b=a

In [3]: c = a[:]

In [4]: a==b
Out[4]: True

In [5]: a is b
Out[5]: True

In [6]: a == c
Out[6]: True

In [7]: a is c
Out[7]: False

In [8]:
```

Else If

```
In [11]: if 4 in a[:]:
    ...:     print("In the list")
    ...: else:
    ...:     print("Not in the list")
    ...:
In the list

In [12]:
```

CS620
HW1
@Ashish_Verma_and_01266453

Loops

```
In [12]: for x in range(10):
    ...:     if x == 3:
    ...:         continue
    ...:     if x == 5:
    ...:         break
    ...:     print(x)
    ...:
0
1
2
4

In [13]:
```

Functions

```
In [13]: items = [1,3,5,7,9]

In [14]: updated_items = list(map(lambda x : x -1,items))

In [15]: updated_items
Out[15]: [0, 2, 4, 6, 8]

In [16]:
```

Functions as Objects

CS620
HW1
@Ashish_Verma_and_01266453

```
In [16]: def pow(x):
    ...:     print(x  ** 2)
    ...:

In [17]: power = pow

In [18]: power(3)
9

In [19]:
```

Range Function

```
In [19]: print(range(5))
range(0, 5)

In [20]: print(list(range(6))
    ...:
    ...:
    ...: )
[0, 1, 2, 3, 4, 5]

In [21]:
```

```
In [21]: for i in range(10,2,-2):
    ...:     print(i)
    ...:
10
8
6
4

In [22]:
```

CS620
HW1
@Ashish_Verma_and_01266453

Activity 1

The Company Naukha's new CEO's salary is $350,000. Use the following equation to generate the tax amount, where S=$300,001 is the Start of the income bracket, R=32% is the tax rate and A=$60,400 is the total tax amount owes from previous brackets.

tax = (income - S) * R + A
tax = (350,000 – 300,001) x 32% + 60,400 = $76,399.68

- Use the python multiple assignment to assign values for S, R and A.

- Write a simple function calculate_tax that takes the salary as the input and returns the calculated tax value.

```
def calc_tax(salary:int) -> float:

    S,R,A = 300001,0.32,60400

    tax = (salary - S) * R + A

    return tax

calc_tax(350000)

Out[23]: 76399.68
```

List comprehension

CS620
HW1
@Ashish_Verma_and_01266453

```
IPython  -- An enhanced Interactive Python.

In [1]: squares =[]

In [2]: for x in range(5):
   ...:     squares.append(x*x)
   ...: print(squares)
[0, 1, 4, 9, 16]

In [3]: list_squares =[x**2 for x in range(5)]

In [4]: print(list_squares)
[0, 1, 4, 9, 16]

In [5]: odd_numbers = [x for x in range(10) if x %2 ==1]

In [6]: print(odd_numbers)
[1, 3, 5, 7, 9]

In [7]:
```

High Order Functions

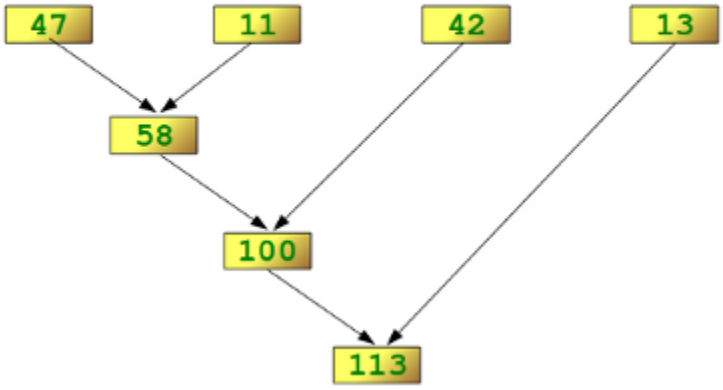
map(function_to_apply, list_of_inputs)

```
In [12]: items =[1,2,3,4,5]

In [13]: squared = list(map(lambda x: x**2, items))

In [14]: print(squared)
[1, 4, 9, 16, 25]

In [15]:
```

The function reduce(func, seq) continually applies the function func() to the sequence seq. It returns a single value.

CS620
HW1
@Ashish_Verma_and_01266453

```
reduce(lambda x,y: x+y, [47,11,42,13])
#output : 113
```



Determining the maximum of a list of numerical values by using reduce and

Calculating the sum of the numbers from 1 to 100:

```
In [19]: func = lambda a,b: a if a>b else b

In [20]: from functools import reduce

In [21]: reduce(func,[47,11,42,102,13])
Out[21]: 102

In [22]: reduce(lambda x,y: x + y, range(1,101))
Out[22]: 5050

In [23]:
```

Filter creates a list of elements for which a function returns true.

CS620
HW1
@Ashish_Verma_and_01266453

```
In [23]: number_list = range(-5,5)

In [24]: less_than_zero = list(filter(lambda x:x<0, number_list))

In [25]: print(less_than_zero)
[-5, -4, -3, -2, -1]

In [26]:
```

Packing/Unpacking

ZIP: Useful to combined multiple lists into a list of tuples

```
In [29]: list(zip([1,2,3,4],[-1,0],[8,8,9]))
Out[29]: [(1, -1, 8), (2, 0, 8)]

In [30]: names =['James','Tom','Mary']

In [31]: grades = [100,90,95]

In [32]: list(zip(names,grades))
Out[32]: [('James', 100), ('Tom', 90), ('Mary', 95)]

In [33]: gradeBookks =[['James',100],['Tom',90],['Mary',95]]

In [34]: [names,grade]= zip(*gradeBookks)

In [35]: print(names)
('James', 'Tom', 'Mary')

In [36]: print(grades)
[100, 90, 95]

In [37]:
```

Module random

CS620
HW1
@Ashish_Verma_and_01266453

```
In [37]: import random

In [38]: rands = [random.randint(0,9) for _ in range(10)]

In [39]: rands
Out[39]: [9, 6, 1, 9, 0, 2, 0, 2, 7, 4]

In [40]:
```