

Summary of Learning Objectives – ENGR 16100
Weeks 6 - 11 : Eligible for assessment on Exam #2

Link to Exam 1 Learning Objectives:

<https://docs.google.com/document/d/1Lfuhl4vCleHE5QbjU7Pe77f-Yo6Zfrra3EMKRWpCqA/edit?usp=sharing>

Link to equation sheet:

https://mycourses.purdue.edu/bbcswebdav/pid-10115616-dt-content-rid-44720959_1/courses/1810W_stanchfi_1/previously_derived_equations_v5.pdf

Any Learning Objective from Exam 1 could appear on Exam 2, though Exam 2 will concentrate on the Learning Objectives listed here.

Link to Quizlet for Midterm #2 that includes some of the main things from Exam 1

https://quizlet.com/_3ze6bi

Force and Motion

Generate a model to analyze the forces acting on a system in motion in either rectilinear or polar coordinates

- $\vec{F} = m\vec{a}$

Use polar /cylindrical coordinates to evaluate kinematic characteristics (position, velocity and acceleration) of a particle in motion

Cartesian	Polar	
$\vec{r} = x\hat{i} + y\hat{j} + z\hat{k}$	$\vec{r} = r\hat{e}_r + z\hat{e}_z$	$\hat{e}_r = \cos\theta\hat{i} + \sin\theta\hat{j}$
$\vec{v} = \dot{x}\hat{i} + \dot{y}\hat{j} + \dot{z}\hat{k}$	$\vec{v} = \dot{r}\hat{e}_r + r\dot{\theta}\hat{e}_\theta + \dot{z}\hat{e}_z$	$\hat{e}_\theta = -\sin\theta\hat{i} + \cos\theta\hat{j}$
$\vec{a} = \ddot{x}\hat{i} + \ddot{y}\hat{j} + \ddot{z}\hat{k}$	$\vec{a} = (\ddot{r} - r\dot{\theta}^2)\hat{e}_r + (r\ddot{\theta} + 2\dot{r}\dot{\theta})\hat{e}_\theta + \ddot{z}\hat{e}_z$	$\hat{e}_z = \hat{k}$

-

Use polar/cylindrical coordinates to determine values for physical properties of a particle system (force magnitudes, masses, friction coefficients, spring constants, etc)

-

Determine the rate of change of momentum in a particle traveling along a non-linear path

- $\Sigma \vec{F} = m\vec{a} = m \frac{d\vec{v}}{dt} = \frac{d\vec{p}}{dt}$

Work and Energy

Determine for a given situation whether or not a certain force is doing work on a particle in motion

- In order to accomplish work on an object there must be a force exerted on the object and it must move in a direction parallel to the force or a component of the force.

Calculate the work done by a constant force acting on a particle in motion

- $W = \vec{F} \cdot \vec{d}$

Calculate the kinetic energy of a particle in motion

- Kinetic: $E_k = \frac{1}{2} mv^2$

Calculate the gravitational potential energy and/or elastic potential energy of a particle system

- Gravitational Potential: $E_g = mgh$
- Elastic Potential: $E_{sp} = \frac{1}{2} k(\Delta x)^2$

Conservation and Universal Accounting Equation

Define the relationships and distinctions between a state quantity and a path quantity

- State quantity: independent of the path (like energy)
- Path quantity: dependent on the path (like work)

Define the terms of the universal accounting equation

- Final Amount-Initial Amount = Input - Output + Generation - Consumption
 - Final Amount - Specifies the amount of the counted quantity in the system at the end of the time period.
 - Initial Amount - Specifies the amount of the counted quantity in the system at the beginning of the time period.
 - Input - Specifies the amount of the counted quantity passing through the boundary into the system during the time interval.
 - Output - Specifies the amount of the counted quantity passing through the boundary out of the system during the time interval.
 - Generation - Specifies the amount of the counted quantity produced during the time interval within the system boundary.
 - Consumption - Specifies the amount of the counted quantity destroyed during the time interval within the system boundary.

Perform a mass and/or energy balance across one or more subsystems to evaluate how a system changes with time

- Gravitational Potential: $E_g = mgh$
- Elastic Potential: $E_{sp} = \frac{1}{2} k(\Delta x)^2$
- Kinetic: $E_k = \frac{1}{2} mv^2$

Identify and explain the major considerations an engineer must maintain when using the universal accounting equation (known boundary, start and end time, accounting on extensive properties)

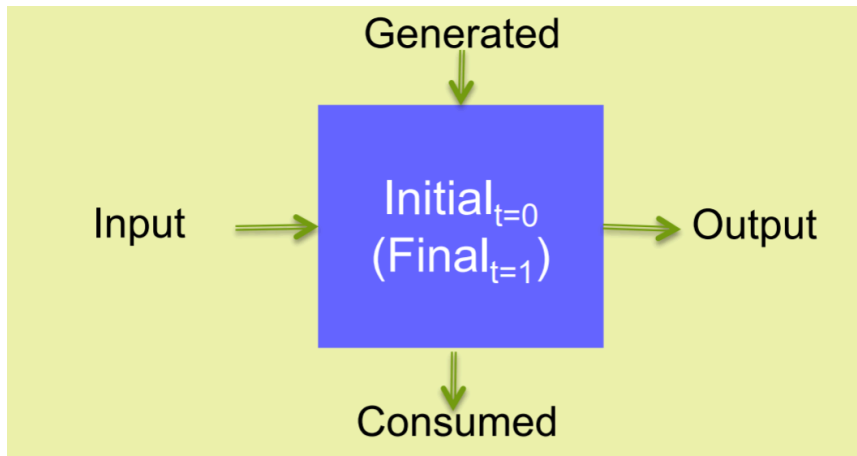
- Define a boundary - Definable boundary, boundary can change over time, whenever something crosses the system boundary it must be accounted for
- Determine what will be counted (eg, energy)
- Determine a time interval for which the accounting will be performed (initial state and end state)

Define the difference between energy, work and power

- Work - when one applies a force to a particle that causes the particle to move through some distance parallel to the force, a force causing a displacement d

- A system has energy if it has the ability to do work
- Power - the rate at which work is done or energy is transferred over time. $P=W/t$, $P=F*v$

Create a system representation diagram



Developing computational (mathematical) models to evaluate an overall system or optimize a given aspect of the system

- Best of luck on this one

Engineering Innovation Methods

Identify, describe, and give examples relating to sources of brainstorming inspiration (nature, prior art, random input)

- Prior Art (looks to solutions to other problems to come up with solutions for the given problem)
- Nature (look at the way nature accomplishes a task, identify how the function to be accomplished has been done before)
- Random Entry Method (think about novel associations of ideas by focusing on something completely unrelated)

Identify, describe, and give examples relating to methods for brainstorming discussed in class (plain brainstorming, random entry method, Brainwriting, and 'The Worst')

- Random Entry Method (think about novel associations of ideas by focusing on something completely unrelated)
- Brainwriting (each participant writes down three ideas as fast as they can and passes the paper to the next person who writes 3 new ideas)
- The worst possible (create the worst possible design to ensure creative thought)
- Brainstorming by analogy (Prior Art, Nature (see above))
- Traditional Brainstorming (common, unsupported creativity technique, usually done by groups, but can be done by individuals, use the rules)

Discuss how to most effectively work with other people on brainstorming tasks, including the 'rules' of brainstorming and how individual and group work should alternate

- The rules: focus on quantity, withhold criticism, welcome unusual ideas, combine and improve ideas

- Groups generate more ideas, more different ideas, more thoughts on how to combine or improve ideas, and have a much better chance of avoiding fixation, but group brainstorming must also be balanced out with individual brainstorming

Python (Exam 1 LO's are NOT listed here but may be included on the Exam)

Predict the output of a complete for and while loop in Python

Algorithm		Trace Table			
		Line	number	i	OUTPUT
1	number = 3	1	3		
2	PRINT number	2			3
3	FOR i, from 1 to 3:	3		1	
4	number = number + 5	4	8		
5	PRINT number	5			8
6	PRINT "2"	3		2	
		4	13		
		5			13
		3		3	
		4	18		
		5			18
		6			?

- Use trace tables

Use 'for' loops to conduct repetitive operations in Python

- for x in list:
- [float(i) for i in list]
- For x in range(0,7):

Use while loops to conduct repetitive operations in Python

- while True: #loops while True is True; i.e. forever
- doSomething()
- break #ends the loop

Explain the engineering applications of loops and how loops can simplify difficult programming tasks in Python

- for the sake of efficiency and our sanity, we can create a loop that automatically executes a block of code until a specified condition is met.

Use loops to create corresponding data sets (lists and one - dimensional arrays) in Python

- mylist = []
- for x in range(1,10):
- mylist.append(x)
- print(mylist) #outputs [1, 2, 3, 4, 5, 6, 7, 8, 9]

Create, manipulate, and read from arrays and lists in Python

Use loops to compare and evaluate associated elements in data sets in Python

- for x in xlist:
- for y in ylist:

- `doSomething(x,y)` #will do something to every possible combination of values in xlist and values in ylist

Use the range command to create lists in Python

- `mylist = list(range(1,10))` #be sure to cast as a list or the next line will print “range(1,10)”
- `print(mylist)` #outputs [1, 2, 3, 4, 5, 6, 7, 8, 9]

Utilize file I/O functions in Python (`open()`, `close()`, `read()`, `readline()`, `readlines()`, `write()`, `writelines()`) to read from and write to files

<https://www.youtube.com/watch?v=Uh2ebFW8OYM>

- `open()`
 - `fid = open(filename, mode)`
 - content generated from this operation.
 - 'a': append - any data written to the file will be added to the end of the existing file.
 - 'r+': read and write - the user can both read and write to the file.
 - 'b': when appended to the mode, opens a file in binary mode
 - Note: If no mode is specified, 'r' is assumed.
 - 'r': read - the file will only be read
 - 'w': write - for writing only. Any file with the same name will be replaced by the
- `.close()`
 - Once you are finished reading or writing to a file, you must close the file. When you close a file, you no longer use system resources. The close operation is
 - `filename.close()`
 - If you don't close a file, eventually Python's garbage service will close the open file and destroy the file object. However, until this happens, the file remains open.
- `.read()`
 - `filename.read(size)`
 - In this operation, if size is:
 - a positive integer: a specified quantity of content is returned. This corresponds to the number of characters for a text file or the number of bytes in a binary file.
 - a negative integer: all file contents are imported
 - omitted: all file contents are imported
- `.readline()`
 - This operation reads in file content line by line
 - Omitting the number of characters or bytes will result in the import of the full line of text.
 - When the user entered the command `exercise.readline()` for the first time, the entire first line of the file was imported. When the user entered the command again, but specified a 4 inside the parentheses:
`exercise.readline(4)`

the first 4 characters of the second line were imported.

- `.readlines()`
 - This operation reads in file content lines in a list format, including the “\n” newline character in each element
- `.write()`
 - write mode: `fid = open(filepath, 'w')`
 - append mode: `fid = open(filepath, 'a')`
 - binary mode: `fid = open(filepath, 'wb')` or `fid = open(filepath, 'ab')`
- `.writelines()`
 - add the \n to create a new line in the content you are exporting.
 - `fileObject.writelines( sequence )`
 - ?
 - sequence – This is the Sequence of the strings

Construct files using formatted output using Python

- <https://docs.python.org/3/tutorial/inputoutput.html>

Read in input data from a user at the keyboard in Python

- `name = input("What is your name?")`
- `print(name)`
- #Outputs 'What is your name?', then waits for user input
- #User i types 'Sir Robin' and presses enter; outputs 'Sir Robin'

Generate basic plots in two dimensions with appropriate axis, title and legend identifiers

Plot multiple set of data on a single plot

- **This is an example of a plot developed from reading in files**

```
import matplotlib.pyplot as plt
```

```
x_text = open('x.txt', 'r')
x_text_raw = x_text.readlines()
x = x_text_raw
```

```
y_text = open('y.txt', 'r')
y_text_raw = y_text.readlines()
y = y_text_raw
```

```
plt.figure(1)
plt.plot(x,y, color='aquamarine', linewidth=9, linestyle=':', marker='*', ms=4,mfc='red')
plt.title('Projectile Motion Plot', fontsize=15)
plt.ylabel('height [meters]', fontsize=15)
plt.xlabel('distance [meters]', fontsize=15)
plt.grid()
```

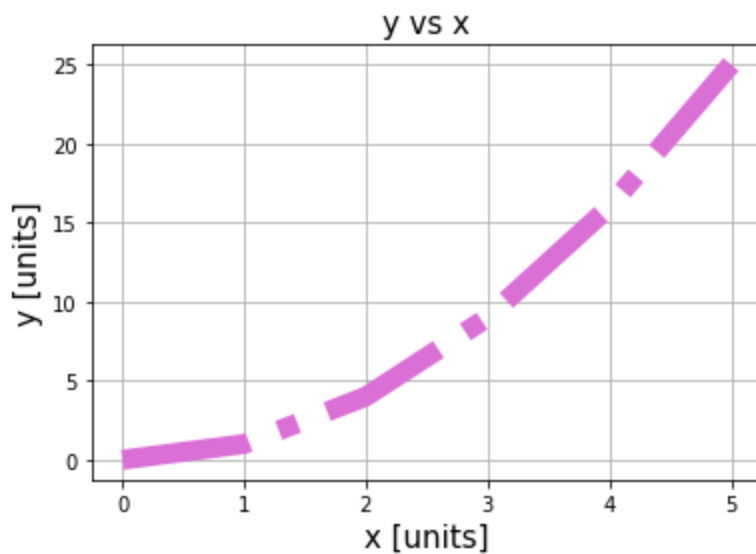
This is an example of a plot using a user developed list

```
import matplotlib.pyplot as plt

x = [0,1,2,3,4,5]
y = [0,1,4,9,16,25]

plt.figure(1)
plt.plot(x,y, color='orchid', linewidth=10, linestyle='-.')
plt.title('y vs x', fontsize=15)
plt.ylabel('y [units]', fontsize=15)
plt.xlabel('x [units]', fontsize=15)
plt.grid()
```

Output:



Generate subplots on a single figure

Previous Exam Learning Objectives of Special Note

Using the Problem Presentation Method to systematically solve a problem

- **PPM:**
 - Given is not needed on exam
 - Find
 - be specific
 - Representation
 - list measurements as variables
 - only use given numbers
 - *always* draw an FBD



- Include axes
- Model
 - equations have no numbers unless a constant or conversion factors
- Assumptions
 - include anything the problem doesn't give you
 - ideal conditions
- Solution
 - include units
 - number steps in coordination with model
 - box final answer
 - not required for symbolic problems
- Validation
 - Set Bounds
 - Solve with different model
 - not required for symbolic problems

Flowcharting

- Construct flowcharts using paper and pencil or digital tools for sequential, conditional, and repetitive logical structures.

Process Type	Representation	Graphic Symbol
input/output	parallelogram	
process instruction	rectangle	
line connectors	Line with arrowhead indication flow direction	
conditional test or decision	diamond	
start/stop	oval	
connectors – on page off page	circle pentagon	
subroutine	divided rectangle	

- Conditional Flow - A path is chosen based on whether a condition is met or not (yes or no/ true or false, etc.) Like a fork in the road
- Repetition Flow - Looping of commands until a condition is met
- Sequential Flow - Series of statements conducted in order written. No decisions

Basics of engineering design, teaming and project management tools

- Explain defining features of engineering design.
 - A systematic, intelligent process in which designers generate, evaluate, and specify concepts for device, system or processes whose form and function

achieve client's' objectives or users needs while satisfying a specified set of constraints.

- Define Iteration
 - Going back to a previous step in the design process and redoing it to achieve better results
 - Define Constraint
 - A limiting factor or resource of the problem
 - Define Stakeholder
 - A client or anyone who has an interest or will be affected by the final design of the product
 - Define Goal
 - An informed statement of what the end product should be
- List and describe the steps in a typical engineering design process
 - **Identify opportunity**
 - Notice opportunities for change
 - Articulate goals
 - **Define Problem**
 - Represent the system's functions and relationships
 - Identify factors influencing design
 - Articulate some problems to be solved
 - Generate questions about things you need to know
 - **Acquire and Assemble the Pertinent Data**
 - Evaluate current solutions on the market
 - Seek additional information to further define problem space
 - Answer questions and inform idea creation
 - **Identify Solution's Constraints and Criteria**
 - Define measures of success (metrics)
 - Order the list of requirements to manage complexity
 - **Develop Alternative Solutions**
 - Brainstorm ideas
 - List and organize
 - Model ideas and anticipate potential outcomes
 - Compare and contrast ideas relevant to requirements
 - **Select a Solution based on Analysis of Alternatives**
 - Evaluate metrics
 - Consider relative importance of criteria
 - Build models or prototypes
 - Refine the design
 - **Communicate Results**
 - Documentation
 - Drawings and construction information
 - Record of evidence

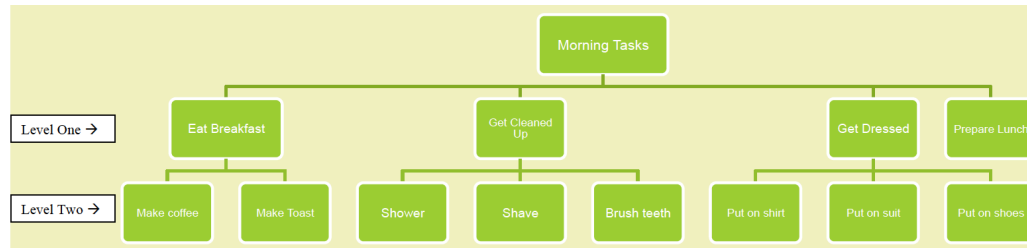
- Approvals
- Define and distinguish the project's goals, customer needs, technical needs, technical requirements and targets.
 - Customer needs - what customer wants ex. Cost effective
 - Technical Needs - units (how we're going to measure it) ex. \$
 - Technical Requirements - numbers ex. Less than \$300
 - Target Values - measurement goal ex. \$250+ or - \$50
- Generate measurable engineering specifications involving technical needs, technical requirements and target values for familiar "systems" (IE toast machine (toaster?), ice maker, or similar).
- Rules of Brainstorming
 1. Focus on quantity
 - The greater the number of ideas generated, the greater the chance of producing a radical and effective solution
 2. Withhold criticism
 - So that participants will feel free to generate unusual ideas
 - Criticism is done in a later stage of the process
 3. Welcome unusual ideas
 - Be willing to look at things from new perspectives or without certain assumptions
 4. Combine and improve idea
 - Good ideas combined to form a single better idea ("1+1=3")
 - Stimulates the building of ideas by a process of association
- Define key characteristics of a team, as opposed to a group.

"A team is a small group of people with **complementary skills** who are **committed** to a **common purpose, performance goals**, and **approach** for which they hold themselves **mutually accountable**"
- Describe the main issues that influence team effectiveness.
 1. Lack of participation or attendance
 2. Lack of organization
 3. Losing track of time
 4. Lack of communication
 5. Cohesiveness, Communication, Roles and Norms, Interdependence, Goal
- Describe the stages of team development.
 1. Forming (orientation) - Tentative interactions; polite discourse; concern over role ambiguity; and self discourse
 2. Storming (conflict) - Criticism of ideas; poor attendance; hostility; polarization; and coalition forming.

3. Norming (cohesion) - Agreement on procedures; reduction in role ambiguity; revise Code of Cooperation based upon current experiences; and increased "we feeling"
 4. Performing (performance) - Decision making; problem solving; mutual cooperation; high task orientation; and emphasis is placed upon performance and production.
 5. Adjourning (dissolution) - Team is dissolved
- Describe the team roles suggested for this class (meeting coordinator, encourager, etc.)
 1. Meeting Coordinator
 - Coordinates and prepares the agenda (i.e, what needs to be accomplished, establishes a process, etc.);
 - coordinates time, date and place of meetings;
 - ensures all necessary resources are available for the meetings;
 - monitors the decision making process;
 - coordinates the check for understanding.
 - (keeper of the Code of Cooperation)
 2. Time Keeper
 - Keeps track of time
 - Keeps the team on task so that they can finish the activity at hand
 - Monitors the agenda and allocated time for discussion – calls team for process check
 3. Recorder
 - Does the writing of the team whenever group work is being done.
 - Maximizes participation by the rest of the team, (no one else needs to worry about it)
 - If required, also ensures that the processes being used are documented and/or prepares an ACTION LIST to keep a record of assigned actions.
 - Makes sure that copies of their work are provided to the rest of the team.
 4. Encourager / Gatekeeper
 - Has the task of giving encouragement to all the other team members.
 - Also has the responsibility of maintaining a balanced level of participation for all the members.
 - They will encourage the silent members
 - They will try to hold back the verbose, dominant members.
 5. Devil's Advocate
 - Takes a position opposite to that held by the team to ensure that all sides of an issue are considered.
 - This responsibility should be undertaken by all team members.
 - Identify and describe the use of specific strategies for solving team problems (team roles, code of cooperation, effective meetings, etc.)
 1. **Code of Cooperation**- the agreed upon rules governing the behavior of team members, as well as any appropriate rewards and sanctions. It sets a norm for acceptable behavior for each team member and represents how the team

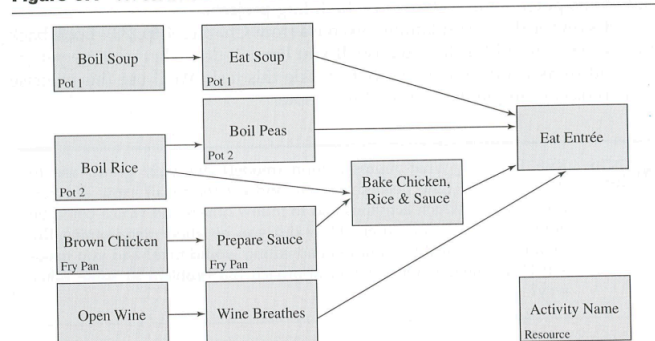
members will interact with one another. It should be developed, adopted, improved and/or modified by all team members on a continuous basis. It should be easily accessible to team members.

- **Individual behavior rules-** the team expectation of each member's individual behavior/contribution to the team
- **Overall team behavior rules-** everyone's expectation of team behavior/contribution
- **Develop Work Breakdown Structures (WBS).**
 - Provide a framework: scheduling, assigning responsibilities, cost estimating
 - Visually divides project into manageable chunks
 - Has a hierarchical structure
 - Document 100% of deliverables
 - Sum of all children = 100% of the parent
 - Focus on what not how
 - Should not have methods or actions



- **Create a precedence network (diagram) from a WBS or from scratch.**
 - Illustrates major activities to achieve and relationship between these actions
 - Helps create a project schedule
 - Identifies missing activities, **dependencies** (activities that depend on another finishing first), **critical** activities

Figure 6.1 A Post-It Note Precedence Network



- **Develop Gantt charts for given processes or projects.**
 - Combination of WBS and Precedence Diagram that shows work breakdown and **schedule** for completing tasks and subtasks
- **Determine a critical path in a Precedence Diagram or Gantt Chart and provide qualitative or numerical evidence to justify the path selected.**

- Critical Path Method (CPM):
 - Total duration of project
 - Range of start times for activities and range of end times for activities
 - Gives you a way to see which activities can be done in parallel
 - Allows you to plan order of accomplishment
 - Activities on the critical path
 - Have zero range for start time and end time.
 - These activities must be done on time, or the entire schedule gets delayed!
 - These activities usually need special attention/planning in order for entire project to succeed.

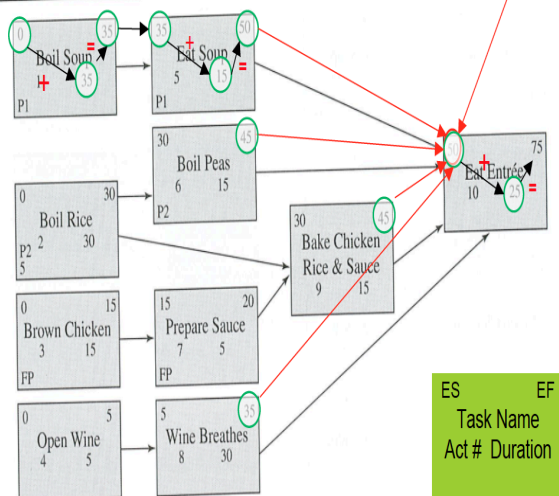
Early Finish

Late Finish

Compute longest time for

How is this value determined?
 $50 > 45 > 35$

Figure 6.3 CPM: Forward Pass—Early Times



Compute Late Times

Figure 6.4 CPM: Backward Pass—Late Times

