MATH 1300-201: EXAM 3 INFO/LOGISTICS/ADVICE

- **INFO:**

WHEN:	Tuesday (07/28) at 8:00am	DURATION:	110 mins
PROBLEM COUNT:	Six	BONUS COUNT:	Two

  - TOPICS CANDIDATE FOR THE EXAM: ("PIRNOT" means the textbook, 5^{th} ed.)
    * PIRNOT 3.1: Converting Logic Statements to/from Symbolic Logic Form, Connectives, Negating Quantifiers
    * PIRNOT 3.2: Truth Tables of AND ($\wedge$), OR ($\vee$), NOT ($\sim$); Logical Equivalence ($\Longleftrightarrow$); DeMorgan's Laws
    * PIRNOT 3.3: Truth Tables of Conditional ($\longrightarrow$), Biconditional ($\longleftrightarrow$); Converses, Inverses, Contrapositives
    * PIRNOT 3.4: Verifying Arguments via Truth Tables, Identifying Common Arguments & Fallacies
    * PIRNOT 4.1: Graphs, Euler Paths, Euler Circuits, Euler's Theorem, Graph Tracing
    * PIRNOT 4.2: Hamilton Paths, Hamilton Circuits
    * PIRNOT 4.2: Weighted Graphs, Solving TSP by Brute Force, Solving TSP by Nearest Neighbor Algorithm
    * PIRNOT 4.3: Directed Graphs, Directed Paths, Incidence Matrices, Modeling Spread of Disease/Information
    * REMARK: **Do not Memorize Formulas – A Formula Sheet will be provided (next two pages)**
  - TOPICS CANDIDATE FOR BONUS QUESTIONS:
    * PIRNOT 3.5: Verifying Quantified Arguments (Syllogisms) via Euler Diagrams
    * ?????
    * REMARK: **Maximum Bonus Points Possible = 30**
  - TOPICS NOT COVERED AT ALL:
    * PIRNOT 3.6: Fuzzy Logic (entire section)
    * PIRNOT 4.1: Using Fleury's Algorithm to find an Euler Path/Circuit, Eulerizing a Graph, Map Coloring
    * PIRNOT 4.2: Solving TSP by the Best Edge Algorithm
    * PIRNOT 4.3: Modeling Influence & Rankings using 1-Stage & 2-Stage Preference
    * PIRNOT 4.4: Scheduling Projects using PERT (entire section)

- **LOGISTICS:**
  - **All you need to bring are pencil(s), eraser(s), calculators(s) & your Raidercard**.
  - Clear your desk of everything except pencil(s), eraser(s), calculator(s).
  - **Backpacks are to placed at the front of the classroom**.
  - **Formula Sheet (next two pages) will be provided**.
  - **Books, notes, notecards NOT PERMITTED**.
  - Mobile devices (phones, tablets, PC's, music, headphones, ...) are to be shut off and put away.
  - Tissues will be furnished – for allergies, not for sobbing.
  - **When you turn in your exam, be prepared to show me your Raidercard if I don't recognize you**.
  - **If you ask to use the restroom during the exam, either hold it or turn in your exam for grading**.

- **ADVICE:**
  - **Use the restroom before the exam, if needed**.
  - **Do not be late – set your wake-up alarms (consider using your cellphone as a backup alarm)**.
  - Review past homework, and perhaps even work some similar problems in the textbook.
  - Review the slides.
  - **Know how the use all formulas on the provided Formula Sheet (next two pages)**
  - **If you need more review, show up to extended office hours on Monday (07/27).**
  - **SHOW APPROPRIATE WORK! Attempt bonus questions**.

MATH 1300: EXAM 3 FORMULA SHEET

PIRNOT 3.1:

- Logic Connectives:

CONNECTIVE NAME:	NOTATION:	MEANING:
Conjunction	$P \wedge Q$	P and Q
Disjunction	$P \vee Q$	P or Q
Negation	$\sim P$	not P
Conditional	$P \longrightarrow Q$	if P then Q
Biconditional	$P \longleftrightarrow Q$	P if and only if Q

- Quantifiers:

  Universal Quantifiers: "All", "Every", "Each"

  Existential Quantifiers: "Some", "At least one", "There exists", "There is/are"

PIRNOT 3.2:

- Truth Tables for Connectives:

P	Q	$P \wedge Q$	$P \vee Q$	$\sim P$	$P \longrightarrow Q$	$P \longleftrightarrow Q$
T	T	T	T	F	T	T
T	F	F	T	F	F	F
F	T	F	T	T	T	F
F	F	F	F	T	T	T

- Logical Connectives (Order of Operations):

DOMINANCE:	CONNECTIVES:
MOST DOMINANT	Biconditional $\longleftrightarrow$
2^{nd} DOMINANT	Conditional $\longrightarrow$
3^{rd} DOMINANT	Conjunction $\wedge$ Disjunction $\vee$
LEAST DOMINANT	Negation $\sim$

- Logical Equivalence: (The symbol $\Longleftrightarrow$ means "is logically equivalent to")

  Two logic statements are **logically equivalent**, if they have the **same variables** (e.g. $P, Q, R, \ldots$) and the **final columns** in their **truth tables** are **identical**.

- DeMorgan's Laws: $\sim(\sim P) \Longleftrightarrow P$, $\quad \sim(P \wedge Q) \Longleftrightarrow (\sim P) \vee (\sim Q)$, $\quad \sim(P \vee Q) \Longleftrightarrow (\sim P) \wedge (\sim Q)$

PIRNOT 3.3:

- Truth Tables for Conditionals & Biconditionals: (see PIRNOT 3.1 above)
- More about Conditionals:

  "If P, then Q" $\Longleftrightarrow$ "Q if P" $\Longleftrightarrow$ "P only if Q" $\Longleftrightarrow$ "P is sufficient for Q" $\Longleftrightarrow$ "Q is necessary for P"

- Converses, Inverses, Contrapositives of Conditionals:

  The **converse** of conditional $P \longrightarrow Q$ is $Q \longrightarrow P$

  The **inverse** of conditional $P \longrightarrow Q$ is $\sim P \longrightarrow \sim Q$

  The **contrapositive** of conditional $P \longrightarrow Q$ is $\sim Q \longrightarrow \sim P$

PIRNOT 3.4:

- Common Arguments & Fallacies:

Law of Detachment:	$P \longrightarrow Q$,	P	$\models Q$
Law of Contraposition:	$P \longrightarrow Q$,	$\sim Q$	$\models \sim P$
Law of Syllogism:	$P \longrightarrow Q$,	$Q \longrightarrow R$	$\models P \longrightarrow R$
Disjunctive Syllogism:	$P \vee Q$,	$\sim P$	$\models Q$
Fallacy of the Converse:	$P \longrightarrow Q$,	Q	$\models P$
Fallacy of the Inverse:	$P \longrightarrow Q$,	$\sim P$	$\models \sim Q$
Affirming a Disjunction:	$P \vee Q$,	P	$\models \sim Q$

PIRNOT 4.1:

- Basics:

  A **graph** consists of a finite set of points, called **vertices**, and lines/curves, called **edges**, that join pairs of vertices.

  An **isolated vertex** has no edges joined to it. A **loop** is an edge which joins one vertex with itself.

  A graph is **connected** if it's possible to travel from any vertex to any other vertex by moving along successive edges.

  An edge of a connected graph is a **bridge** if removing the edge causes the graph to no longer be connected.

- Degree of a Vertex:

  The **degree** of a vertex is the # of edges joined to that vertex.

  Loops count as **two edges**. The degree of an **isolated vertex** is defined to be **zero**.

  An **odd vertex** is a vertex with an odd degree. An **even vertex** is a vertex with an even degree.

- Euler Paths & Euler Circuits:

  A **path** is a sequence of consecutive edges in which no edge is repeated. The **length** of a path is # of edges in the path.

  An **Euler path** is a path that contains all edges of the graph. An **Euler circuit** is an Euler path that begins & ends at same vertex.

- Euler's Theorem:

  A graph can be traced if it's connected and has zero or two odd vertices.

  A graph is traceable if it contains an Euler path or Euler circuit.

  (a) Suppose a graph has two odd vertices.
  Then, the tracing must begin at one odd vertex and end at the other.

  (b) Suppose a graph has zero odd vertices. (i.e. all vertices are even)
  Then, the tracing must begin and end at the same vertex.

PIRNOT 4.2:

- Hamilton Paths & Hamilton Circuits: A **Hamilton path** is a path that passes through all vertices exactly once.

  A **Hamilton circuit** is a Hamilton path that begins & ends at the same vertex.

- Finding an Optimal Hamilton Circuit via Brute Force Algorithm:

  INPUT: A connected weighted graph.

  (1) List all Hamilton circuits in the graph.
  (2) Find the weight of each circuit found in (1).

  OUTPUT: Optimal Hamilton circuit(s) with the smallest weights

- Finding a Semi-Optimal Hamilton Circuit via Nearest Neighbor Algorithm:

  INPUT: A connected weighted graph.

  (1) Start at any vertex.
  (2) Choose the edge joined to the vertex with the smallest weight.
  Traverse the chosen edge.
  (3) Repeat (2) until all vertices have been touched.
  (4) Close the circuit by returning to the starting vertex.

  OUTPUT: Semi-optimal Hamilton circuit with "nearly" smallest weight.

PIRNOT 4.3:

- Directed Edges, Directed Graphs, Directed Paths:

  A **directed edge** is an edge with a direction. A **directed graph** is a graph in which all edges are directed.

  A **directed path** from vertex X to vertex Y in a directed graph is a sequence of edges starting at X, following the edges in the prescribed directions, and ending at Y.

  The **length** of a directed path is the # of edges along that path. The length of a **directed loop** is one (not two).