What is discrete mathematics?

- ➢Discrete mathematics is the part of mathematics devoted to the study of discrete objects.
- ▪ Discrete means consisting of distinct or unconnected elements.
- ✓Distinct – recognizably different in nature from something else of a similar type.
- ➢The kinds of problems solved using discrete mathematics include:
  - How many ways are there to choose a valid password on a computer system?
  - Is there a link between two computers in a network?
  - How can I identify spam e-mail messages?
  - How can a circuit that adds two integers be designed?

Why study Discrete Mathematics?

- Understand and create mathematical arguments
- Discrete mathematics provides the mathematical foundations for many computer science courses including data structures, algorithms, database theory, automata theory, compiler theory, computer security, and operating systems.
- Contains the necessary mathematical background for solving problems in operations research etc.
- Learn mathematical reasoning and problem solving.

1.1 Propositional Logic

TOPIC 1 – The Foundation: Logic and Proofs

The Foundations: Logic and Proofs

- The rules of logic specify the meaning of mathematical statements.
- Logic is the basis of all mathematical reasoning, and of all automated reasoning.
- It has practical applications to the design of computing machines, to the specification of systems, to artificial intelligence, to computer programming, to programming languages, and to other areas of computer science, as well as to many other fields of study.

Propositional Logic

Student Learning Outcomes:

1. Construct correct mathematical arguments,
2. Understand mathematical reasoning, logic in application to computer science,
3. Understand the rules in the design of computer circuits, the construction of computer programs and the verification of the correctness of programs.

Propositions

- **Propositions** – The basic building blocks of logic. A proposition is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.

Example 1 All the following declarative sentences are propositions.

1. Washington, D.C., is the capital of the United States of America.
2. Cebu is the capital of Luzon.
3. 1 + 1 = 2.
4. 2 + 2 = 3.

Propositions 1 and 3 are true, whereas 2 and 4 are false.

Sentences that are not propositions

Example 2 Consider the following sentences.

1. What time is it?
2. Read this carefully.
3. $x + 1 = 2$.
4. $x + y = z$.

Note: Each of sentences 3 and 4 can be turned into a proposition if we assign values to the variables.

Propositional Variables

- We use letters to denote **propositional variables** (or **statement variables),**
- Variables that represent propositions, just as letters are used to denote numerical variables.
- The conventional letters used for propositional variables are p, q, r, s, …. The **truth value** of a proposition is **true**, denoted by **T**, if it is a true proposition.
- The truth value of a proposition is **false**, denoted by **F**, if it is a false proposition.

- The area of logic that deals with propositions is called the **propositional calculus** or **propositional logic**.

- It was first developed systematically by the Greek philosopher Aristotle more than 2300 years ago.

- Many mathematical statements are constructed by combining one or more propositions. New propositions, called **compound propositions**, are formed from existing propositions using logical operators.

Definition 1

Let *p* be a proposition. The *negation of p*, denoted by $\neg p$ (also denoted by $\bar{p}$), is the statement

“ It is not the case that p.”

The proposition $\neg p$ *is read* not p. The truth value of the negation of p, $\neg p$, *is the opposite of the truth value of p*.

Example 3

- Find the negation of the proposition

  "Michael's PC runs Linux"

  and express this in simple English.

Solution: The negation is

"It is not the case that Michael's PC runs Linux."

This negation can be more simply expressed as

"Michael's PC does not run Linux."

Example 4

- Find the negation of the proposition

  "Vandana's smartphone has at least 32GB of memory"

  and express this in simple English.

Solution: The negation is

"It is not the case that Vandana's smartphone has at least 32GB of memory."

"Vandana's smartphone does not have at least 32GB of memory" or even more simply as

"Vandana's smartphone has less than 32GB of memory."

Table 1 – The Truth Table for the Negation of a proposition

p	$\neg p$
T	F
F	T

Table 1 displays the **truth table** for the negation of a proposition p. Each row shows the truth value of $\neg p$ corresponding to the truth value of p for this row.

The **negation operator** constructs a new proposition from a single existing proposition.

Connectives – the logical operators that are used to form new propositions from two or more existing propositions.

Definition 2

- Let p and q be propositions. The *conjunction* of p and q, denoted by $p \wedge q$, is the proposition "p and q." The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.

p	q	$p \wedge q$
T	T	**T**
T	F	**F**
F	T	**F**
F	F	**F**

TABLE 2 The Truth Table for the Conjunction of Two Propositions.

Example 5

- Find the conjunction of the propositions p and q where p is the proposition “ Rebecca’s PC has more than 16 GB free hard disk space” and q is the proposition “The processor in Rebecca’s PC runs faster than 1 GHz.”

Solution:

This conjunction can be expressed more simply as Rebecca’s PC has more than 16 GB free hard disk space, and its processor runs faster than 1 GHz.” For this junction to be true, both conditions given must be true. It is false, when one or both of these conditions are false.

Definition 3

- Let p and q be propositions. The disjunction of p and q, denoted by $p \vee q$, is the proposition “p or q.” The disjunction $p \vee q$ is false when both p and q are false and is true otherwise.

p	q	$p \vee q$
T	T	**T**
T	F	**T**
F	T	**T**
F	F	**F**

TABLE 3 The Truth Table for the Disjunction of Two Propositions.

Example 6

- The use of the connective *or* in a disjunction corresponds to one of the two ways the word *or* is used in English, namely, as an *inclusive or.*
- A disjunction is true when at least one of the two propositions is true.

Example 6: What is the disjunction of the propositions p and q where p and q are the same propositions as in Example 5?

Solution: The disjunction of p and q, $p \vee q$, is the proposition

A disjunction is true when at least one of the two propositions in it is true.

Definition 4

- Let *p* and *q* be propositions. The *exclusive or* of *p* and *q*, denoted by $p \oplus q$, is the proposition that is true when exactly one of *p* and *q* is true and is false otherwise.

p	q	$p \oplus q$
T	T	**F**
T	F	**T**
F	T	**T**
F	F	**F**

TABLE 4 The Truth Table for the Exclusive Or of Two Propositions.

Conditional Statements

- **Definition 5**

Let p and q be propositions. The *conditional statement* $p \rightarrow q$ is the proposition "if p, then q." The conditional statement $p \rightarrow q$ is false when p is true and q is false, and true otherwise. In the conditional statement $p \rightarrow q$, p is called the hypothesis (or antecedent or premise) and q is called the conclusion (or consequence).

TABLE 5 The Truth Table for the Conditional Statement $p \rightarrow q$.

p	q	$p \rightarrow q$
T	T	**T**
T	F	**F**
F	T	**T**
F	F	**T**

Conditional Statements

- The statement $p \rightarrow q$ is called a conditional statement because $p \rightarrow q$ is true on the condition that p holds. A condition statement is also called an **implication**.

- The truth table for the conditional statement $p \rightarrow q$ shown in Table 5. Note that the statement $p \rightarrow q$ is true when both p and q are true and when p is false (no matter what truth value q has).

Conditional Statements

- Because conditional statements play such an essential role in mathematical reasoning, a variety of terminology is used to express $p \to q$. You will encounter most if not all of the following ways to express this conditional statement:

"if p, then q" "p implies q"

"if p, q" "p only if q"

"p is sufficient for q" "a sufficient condition for q is p"

"q if p" "q whenever p"

"q when p" "q is necessary for p"

"a necessary condition for p is q" "q follows from p"

"q unless $\neg\, p$"

Example 7

- Let p be the statement "Maria learns discrete mathematics" and q the statement "Maria will find a good job." Express the statement $p \rightarrow q$ as a statement in English.

Solution: From the definition of conditional statements, we see that when p is the statement "Maria learns discrete mathematics" and q is the statement "Maria will find a good job," p → q represents the statement

"If Maria learns discrete mathematics, then she will find a good job."

There are many other ways to express this conditional statement in English. Among the most natural of these are:

"Maria will find a good job when she learns discrete mathematics."

"For Maria to get a good job, it is sufficient for her to learn discrete mathematics."

Converse, Contrapositive, and Inverse

- Forming some new conditional statements starting with a conditional statement $p \rightarrow q$. In particular, there are three related conditional statements that occur so often that they have special names.
- The proposition $q \rightarrow p$ is called the converse of $p \rightarrow q$.
- The contrapositive of $p \rightarrow q$ is the proposition $\neg q \rightarrow \neg p$.
- The proposition $\neg p \rightarrow \neg q$ is called the inverse of $p \rightarrow q$.
- Of these three conditional statements formed from $p \rightarrow q$, only the contrapositive always has the same truth value as $p \rightarrow q$.
- The inverse is “If it is not raining, then the home team does not win.”

Example

- What are the contrapositive, the converse, and the inverse of the conditional statement "The home team wins whenever it is raining?"
- Solution: Because "q whenever p" is one of the ways to express the conditional statement p → q, the original statement can be rewritten as

"If it is raining, then the home team wins."

Consequently, the contrapositive of this conditional statement is "If the home team does not win, then it is not raining."

The converse is "If the home team wins, then it is raining."

Only the contrapositive is equivalent to the original statement.

BICONDITIONALS

- Another way to combine propositions that expresses that two propositions have the same truth value.
- Let p and q be propositions. The biconditional statement $p \leftrightarrow q$ is the proposition "p if and only if q." The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called bi-implications.
- There are some other common ways to express $p \leftrightarrow q$:

  "p is necessary and sufficient for q"

  "if p then q, and conversely"

  "p iff q."

TABLE 6 The Truth Table for the Biconditional $p \leftrightarrow q$.

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

- The last way of expressing the biconditional statement $p \leftrightarrow q$ uses the abbreviation "iff" for "if and only if." Note that $p \leftrightarrow q$ has exactly the same truth value as $(p \rightarrow q) \land (q \rightarrow p)$.

Example: Biconditionals

- Let p be the statement “You can take the flight,” and let q be the statement “You buy a ticket.” Then $p \leftrightarrow q$ is the statement

  “You can take the flight if and only if you buy a ticket.”

This statement is true if p and q are either both true or both false,

That is, if you buy a ticket and can take the flight or if you do not buy a ticket and you cannot take the flight.

It is false when p and q have opposite truth values.

Truth Tables of Compound Propositions

- Four important logical connectives—conjunctions, disjunctions, conditional statements, and biconditional statements—as well as negations.
- Use these connectives to build up complicated compound propositions involving any number of propositional variables.
- Use a separate column to find the truth value of each compound expression that occurs in the compound proposition as it is built up. The truth values of the compound proposition for each combination of truth values of the propositional variables in it is found in the final column of the table.

Example: Compound Propositions

- Construct the truth table of the compound proposition

$$(p \vee \neg q) \rightarrow (p \wedge q).$$

The truth table of $(p \vee \neg q) \rightarrow (p \wedge q)$

p	**q**	**¬q**	**(p ∨ ¬q)**	**(p ∧ q)**	**(p ∨ ¬q) → (p ∧ q)**
T	T				
T	F				
F	T				
F	F				

Precedence of Logical Operators

- construct compound propositions using the negation operator and the logical operators
- generally use parentheses to specify the order in which logical operators in a compound proposition are to be applied
- For instance, $(p \vee q) \wedge (\neg r)$ is the conjunction of $p \vee q$ and $\neg r$
- to reduce the number of parentheses, we specify that the negation operator is applied before all other logical operators.
- This means that $\neg p \wedge q$ is the conjunction of $\neg p$ and q, namely,$(\neg p) \wedge q$, not the negation of the conjunction of p and q, namely $\neg(p \wedge q)$

TABLE 8
Precedence of Logical Operators.

Operator	*Precedence*
$\neg$	1
$\wedge$ $\vee$	2 3
$\rightarrow$ $\leftrightarrow$	4 5

Precedence of Logical Operations

- Another general rule of precedence is that the conjunction operator takes precedence over the disjunction operator, so that $p \land q \lor r$ means $(p \land q) \lor r$ rather than $p \land (q \lor r)$.
- may be difficult to remember, we will continue to use parentheses so that the order of the disjunction and conjunction operators is clear
- accepted rule that the conditional and biconditional operators $\rightarrow$ and $\leftrightarrow$ have lower precedence than the conjunction and disjunction operators, $\land$ and $\lor$
- Consequently, $p \lor q \rightarrow r$ is the same as $(p \lor q) \rightarrow r$
- use parentheses when the order of the conditional operator and biconditional operator is at issue, although the conditional operator has precedence over the biconditional operator

TABLE 8
Precedence of Logical Operators.

Operator	*Precedence*
$\neg$	1
$\land$ $\lor$	2 3
$\rightarrow$ $\leftrightarrow$	4 5

Logic and Bit Operations

- Computers represent information using bits
- A bit is a symbol with two possible values, namely, 0 (zero) and 1 (one).
- This meaning of the word bit comes from binary digit, because zeros and ones are the digits used in binary representations of numbers.
- The well-known statistician John Tukey introduced this terminology in 1946.
- A bit can be used to represent a truth value, because there are two truth values, namely, true and false

Logic and Bit Operations

- use a 1 bit to represent true and a 0 bit to represent false
- 1 represents T (true), 0 represents F (false)
- A variable is called a Boolean variable if its value is either true or false
- a Boolean variable can be represented using a bit.
- Computer bit operations correspond to the logical connectives.

- By replacing true by a one and false by a zero in the truth tables for the operators ∧, ∨, and ⊕ , the tables shown in Table 9 for the corresponding bit operations are obtained.
- use the notation OR, AND, and XOR for the operators ∨, ∧, and ⊕, as is done in various programming languages.

TABLE 9 Table for the Bit Operators *OR*, *AND*, and *XOR*.

x	y	$x \vee y$	$x \wedge y$	$x \oplus y$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

Logic and Bit Operations

- Definitions: A bit string is a sequence of zero or more bits. The length of this string is the number of bits in the string.

- Example: 101010011 is a bit string of length nine.

- We can extend bit operations to bit strings. We define the bitwise OR, bitwise AND, and bitwise XOR of two strings of the same length to be the strings that have as their bits the OR, AND, and XOR of the corresponding bits in the two strings, respectively.

- We use the symbols $\vee$, $\wedge$, and $\oplus$ to represent the bitwise OR, bitwiseAND, and bitwise XOR operations, respectively. We illustrate bitwise operations on bit strings with Example 13

Example:

- Find the bitwise OR, bitwise AND, and bitwise XOR of the bit strings 01 1011 0110 and 11 0001 1101. (Here, and throughout this book, bit strings will be split into blocks of four bits to make them easier to read.)

- Solution: The bitwise OR, bitwise AND, and bitwise XOR of these strings are obtained by taking the OR, AND, and XOR of the corresponding bits, respectively. This gives us

01 1011 0110
<u>11 0001 1101</u>
11 1011 1111 bitwise OR
01 0001 0100 bitwise AND
10 1010 1011 bitwise XOR