Mathematics for Computing

4COSC007C

Lecture 1

Module Introduction. Types of Numbers. Number Theory Basics.
Modular Arithmetic. Sequences. Introduction to Sets.

Module Organisation

- Module Duration: One Semester, 12 Weeks
- Weekly Lectures (2hrs) & Tutorials (2hrs)
- Attendance is essential

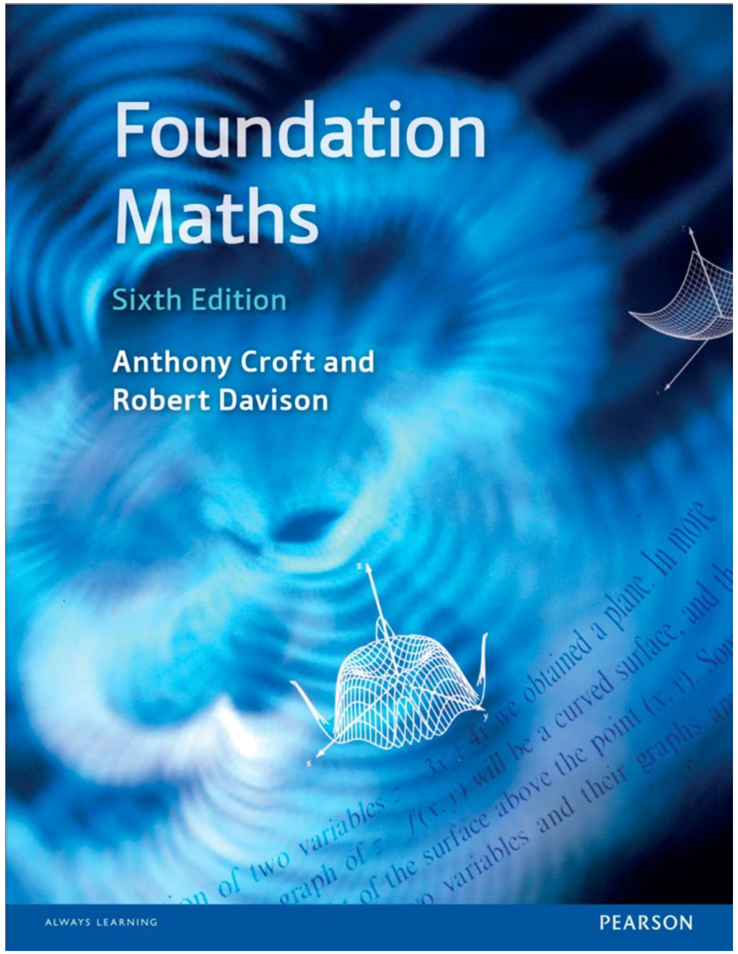
!MODULE ASSESSMENT!

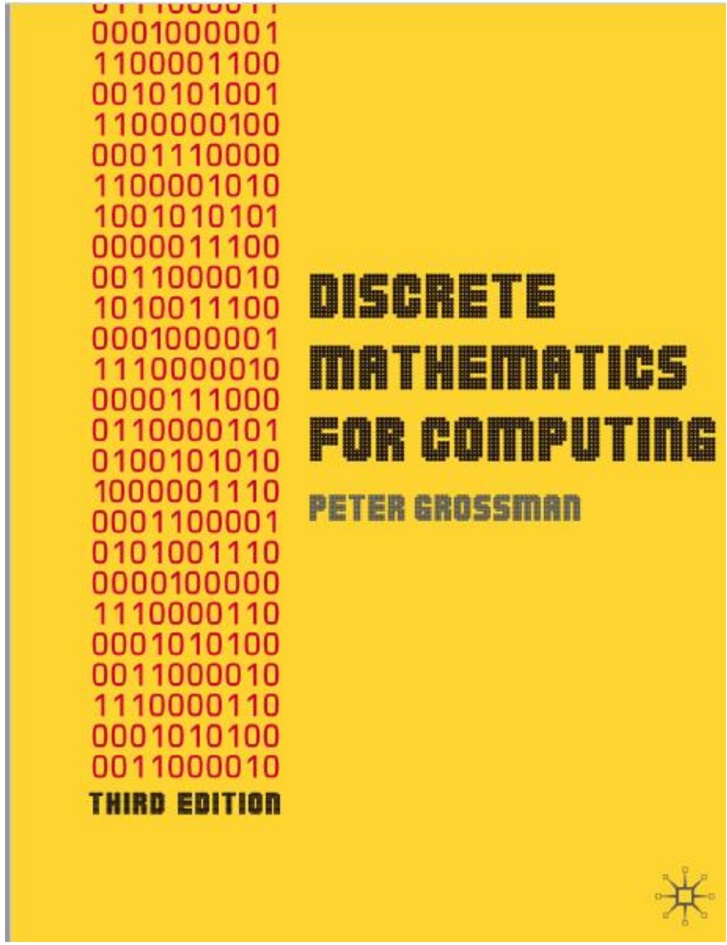
- 2 Assessments: ICT1 & ICT 2. No Coursework.
- **ICT-1** – Week 8. In-Class Test. Duration: 60 minutes. Closed book. Multiple choice questions.
- **ICT-2** – Written Exam. Parts A & B. Part A - Short Answers. Part B – Full Detailed Answers. Duration: 90 minutes
- Overall Module Mark = 50%ICT1+50%ICT2

Module Syllabus

- Number Theory Basics
- Sequences. Set Theory. Intervals
- Relations and Functions
- Logics Basics
- Fundamental Data Structures
- Graph Theory
- Matrices
- Probability Theory
- Elementary Statistics
- Math Related Applications

Sources





Core Textbook Croft & Davison Foundation Maths

Why do Maths for Computing is essential for programming?

- **LOGICAL REASONING:** Maths sharpens the ability to think logically and solve problems, essential for writing efficient algorithms and code.
- **ALGORITHMIC FOUNDATION:** Many algorithms are based on mathematical principles and formulas.
- **DATA STRUCTURES:** Concepts like sets, graphs, and trees, foundational in maths, are vital in creating efficient data structures.
- **MACHINE LEARNING/AI:** Requires understanding of statistics, probability, linear algebra, and calculus.
- **COMPUTER GRAPHICS:** Geometry and trigonometry are fundamental for rendering images and animations.
- **CRYPTOGRAPHY & SECURITY:** Number theory and modular arithmetic form the basis of cryptographic algorithms.
- **EFFICIENCY AND OPTIMIZATION:** Maths helps analyse and improve code efficiency through big O notation and other methods.

ACTIVITY: Home Revision Task

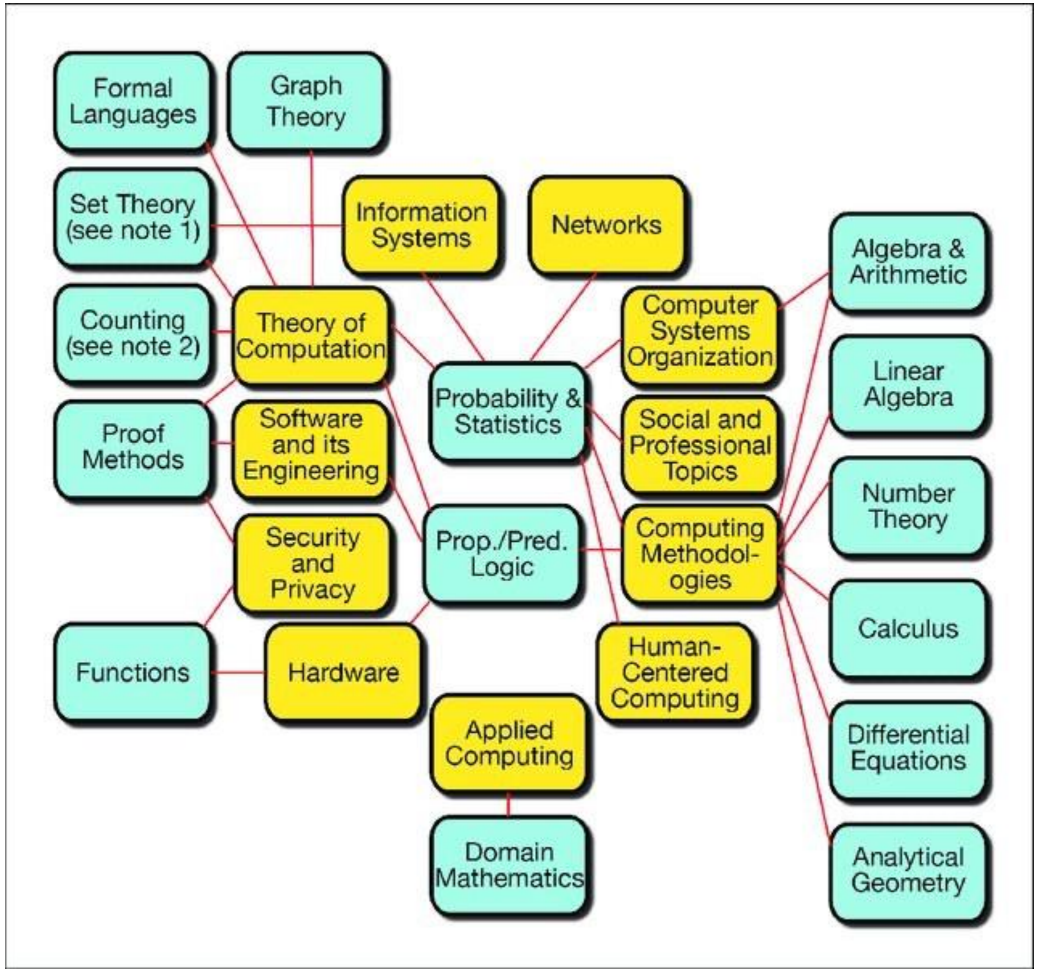
Skim over

CHAPTERS 1-10
Croft & Davison Foundation Maths textbook

to make sure that you know the MATHS BASICS

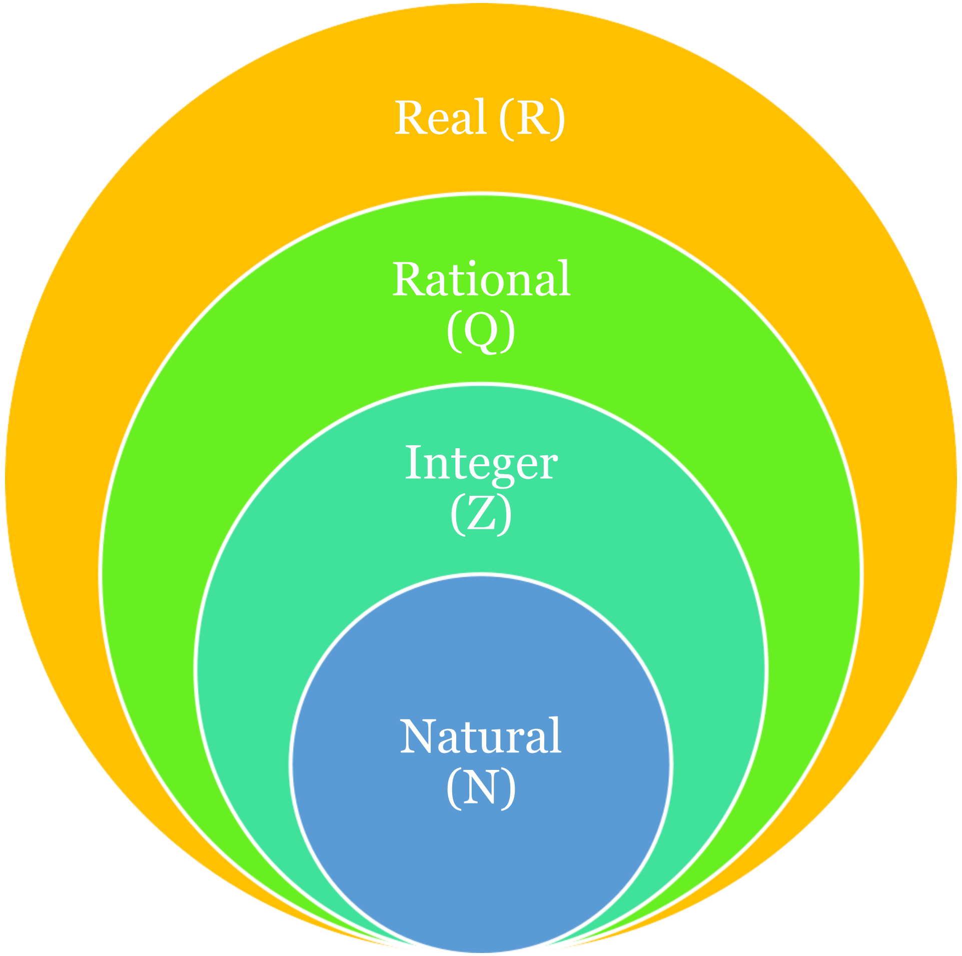
CHAPTERS 1-10
Croft & Davison Foundation Maths textbook

Mathematics associated with computer science areas



Source: Baldwin, D., Walker, H.M. and Henderson, P.B., 2013. The roles of mathematics in computer science. *Acm Inroads*, *4*(4), pp.74-80.

Number Types

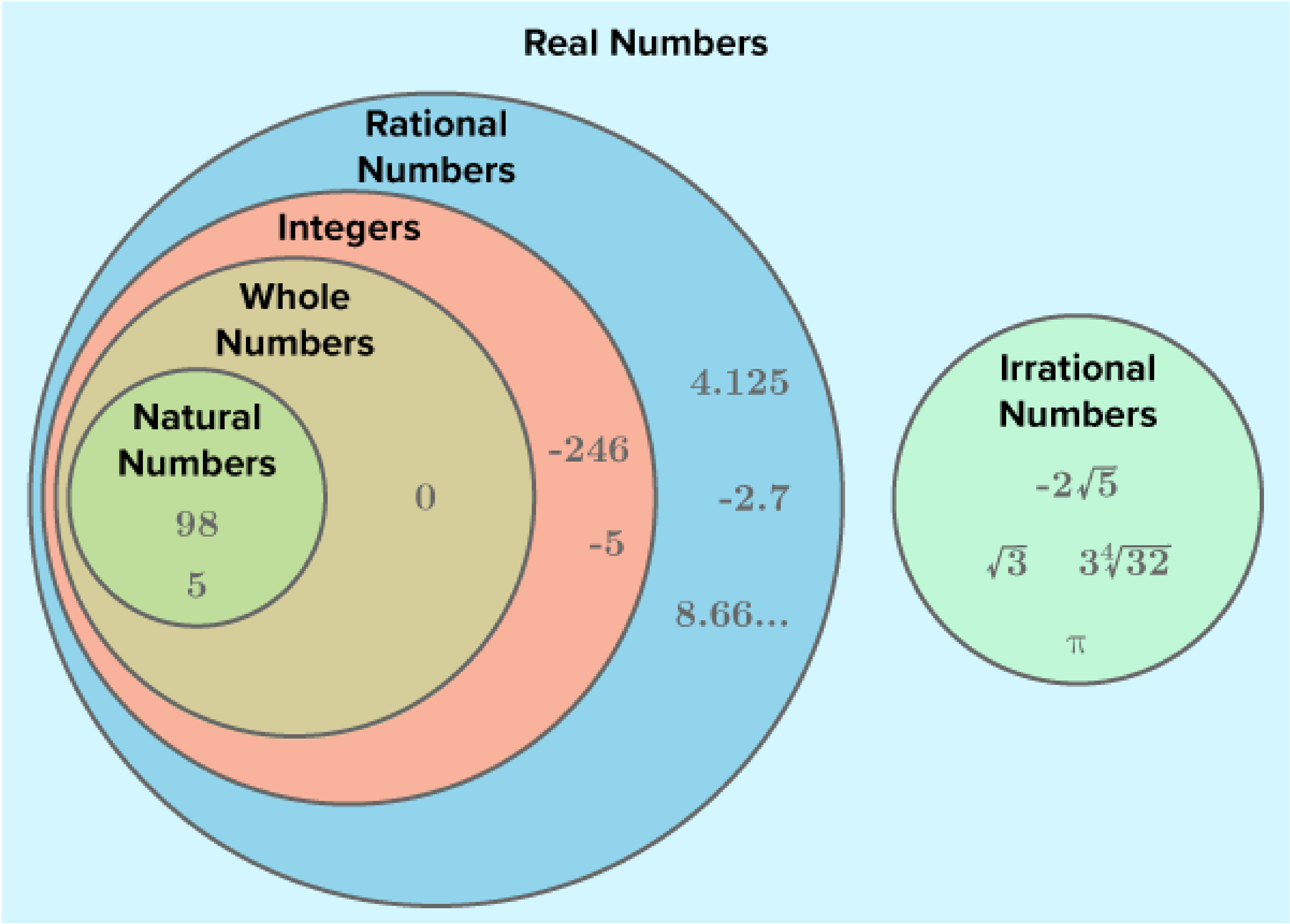


Rational (can be expressed as the quotient or fraction of two integers): ½, ¾, -1,457

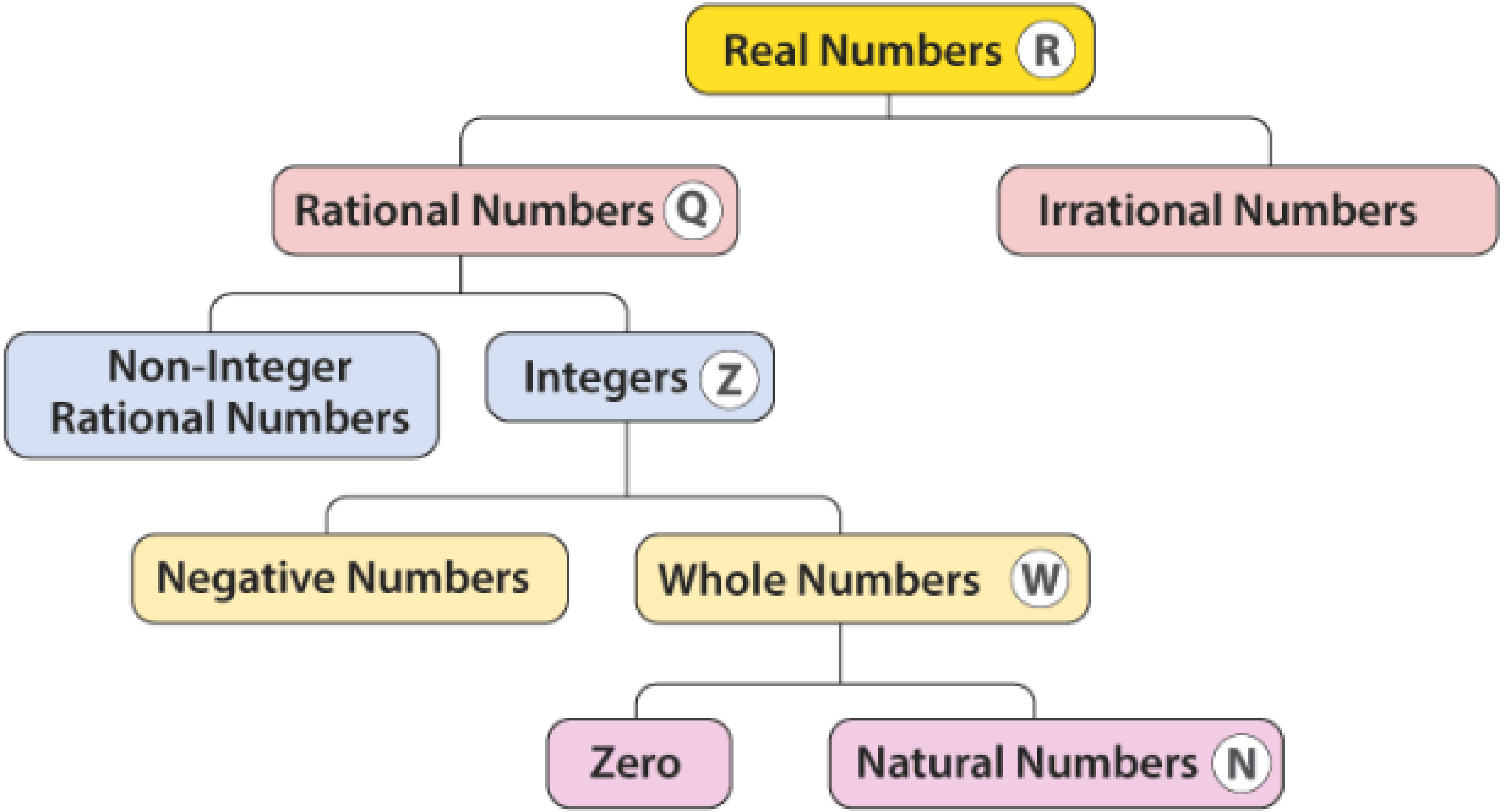
Integers: -3,-2,-1,0,1,2,3

Natural: 1,2,3,4,5

Number Types



Number Types



Source: https://byjus.com/maths/real-numbers/

Introduction to the Number Theory

NUMBER THEORY is a branch of pure mathematics devoted to the study of integers and, more generally, to objects built out of them.

Integers

Set of whole numbers, including positive numbers, negative numbers,

and zero. Notation: **Z** denotes the set of all integers

Prime and Composite Numbers

- **Prime Numbers**: Natural numbers greater than 1 that have no positive divisors other than 1 and itself.
- **Composite Numbers**: Natural numbers greater than 1 that are not prime.

Example: 5 is prime because it can only be divided by 1 and 5, but 4 is composite because it can be divided by 1, 2, and 4.

Fundamental Theorem of Arithmetic

Statement: Every integer greater than 1 is either a prime number itself or can be factorised uniquely into prime numbers up to the order of the factors.

Implication: Prime numbers are the “building blocks” of all natural numbers.

The factorisation is decomposing a number into a product of other numbers.

In the context of the Fundamental Theorem of Arithmetic, factorisation refers specifically to representing an integer as a product of prime numbers.

Example of Factorisation:

Consider the number 84. We can factorise it as follows:

$$84=2\times 42=2\times(2\times 21)=2\times 2\times(3\times 7)=2^2\times 3\times 7$$

Why are Integers popular in coding?

- Basic Data Type
- Memory Efficiency
- Looping and Indexing
- Bitwise Operations

Basic Data Type

- Direct Representation: Integers represent whole numbers without any fractional or decimal part, making them easier to work with in many applications where exact, discrete values are needed.

- No Precision Issues: Unlike floating-point numbers, integers are stored and processed exactly.

- Efficient Storage and Computation: Operations on integers (like addition, subtraction, multiplication, etc.) are typically faster and less resource-intensive than operations on floating-point numbers or more complex data types

Memory Efficiency

- Fixed Size: In many programming languages, integers are stored in a fixed number of bits (e.g., 32-bit or 64-bit), which simplifies memory management.

- No Overhead for Decimal Places: Unlike floating-point types that need additional bits to represent decimal points and precision, integers require fewer resources for storage, especially when you don’t need to store fractions.

Looping and Indexing

- Integers are essential for indexing arrays, lists, and other data structures.

- Integers are commonly used as loop counters in iteration constructs (e.g., for loops). They are ideal for specifying ranges, counting the number of iterations, and controlling the execution flow in loops.

```python
for i in range(10):
    print(i)
```

Division Theorem

For two integers a and b where $b \neq 0$, there always exists unique integers q and r such that $\boldsymbol{a = qb + r}$ and $0 \leq r < |b|$.

Example:

$a = 17$, b=3, we can find $q = 5$ and $r = 2$ so that 17 = 3*5+2

a is called the **dividend**
b is called the **divisor**
q is called the **quotient**
r is called the **remainder**

If $r = 0$, then we say b **divides** a or a is **divisible** by b.

Modular Arithmetic Basics

Modular arithmetic is a system of arithmetic for integers, where numbers "wrap around" after reaching a certain value, known as the modulus.

Given an integer n (the modulus), two integers a and b are said to be congruent modulo *n* if their difference a−b is an integer multiple of n.

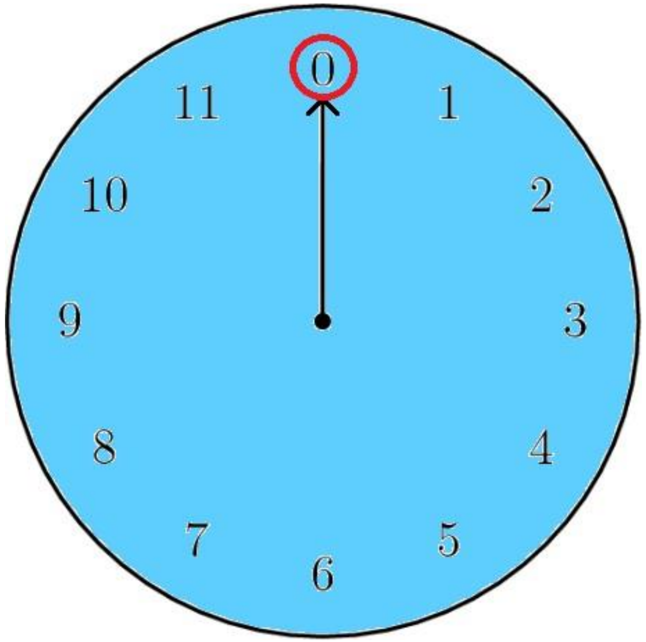
The notation used is: $\boldsymbol{a \equiv b \bmod n}$

Division Theorem: $\frac{A}{B} = \text{Q rem R}$

Example: $\frac{16}{5} = 3 \text{ rem } 1$

16 mod 5 = 1, so A mod B ≡ R

Modular Arithmetic Examples



12hrs Clock

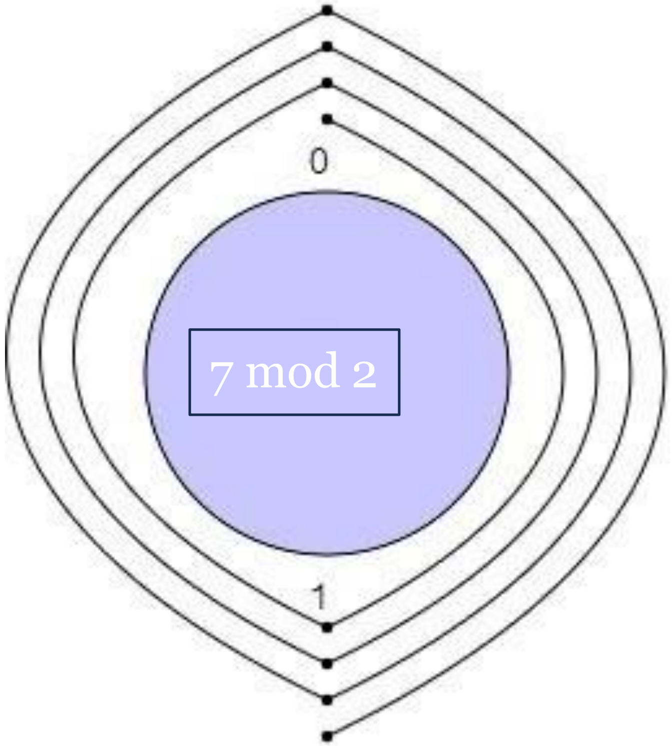
8 hrs +10 hrs =What time is it?
8 hrs +22 hrs=???

U.K. schools getting rid of analogue clocks because teens "cannot tell time."

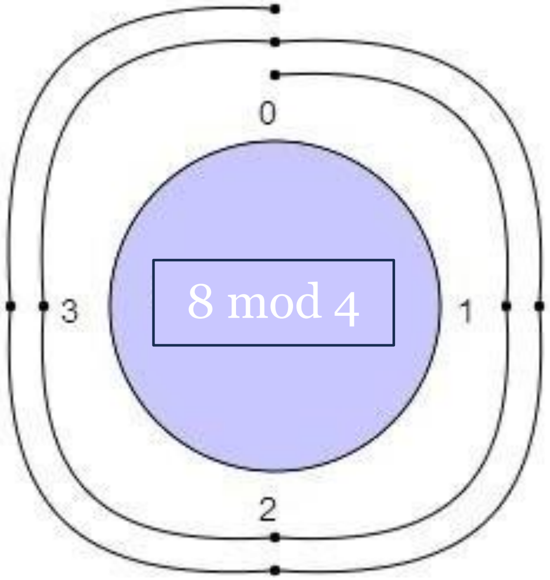
Source: CBS News, May 8, 2018

Modular Arithmetic Examples

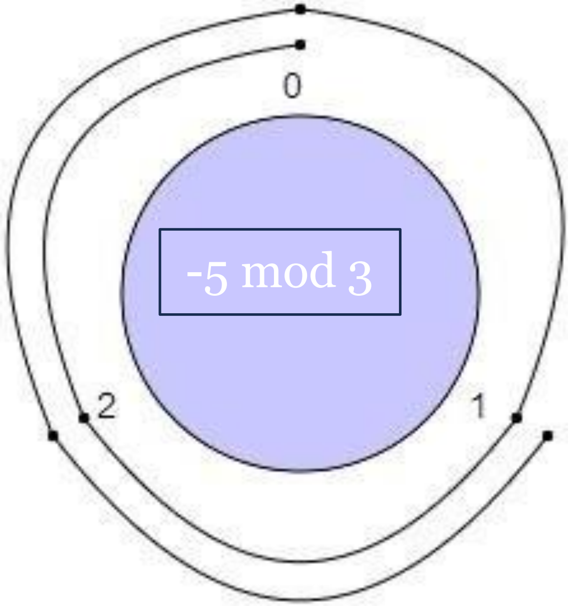
7 mod 2=?



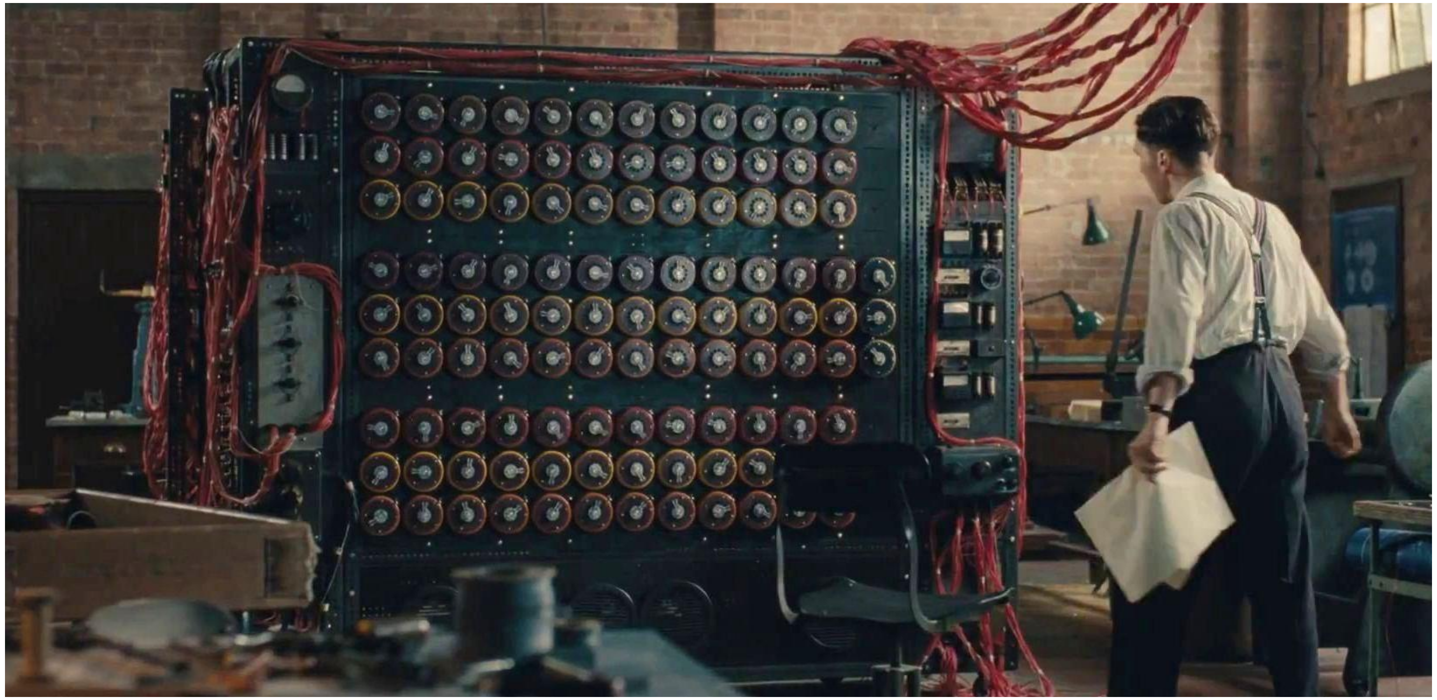
8 mod 4 =?



-5 mod 3 =?



Applications of Modular Mathematics



Source: Imitation Game film screenshot

Cryptography:
Cryptographic systems, including RSA, and Caesar Cipher for encrypting and decrypting messages

Computer Science:
Hashing functions, random number generation, and algorithms for computer graphics.

Sequences

A sequence is an ordered list of elements (numbers, objects, etc.), usually arranged according to a certain rule or pattern.

Each element in a sequence is called **a term**, and the position of a term in a sequence is called its **index**.

Example

(2,4,6,8,10,12) – sequence of even numbers

(1,3,5,7,9,11) – sequence of odd numbers

Sequences can be *finite* or *infinite*.

Arithmetic Sequence

An arithmetic sequence (progression) is a sequence of numbers in which the difference between consecutive terms is constant. This difference is called the "common difference," denoted by *d*. The general form of an arithmetic sequence is

a, a+d, a+2d, a+3d, ..., here, *a* is the first term of the sequence.

The n^{th} term of an arithmetic sequence can be calculated using the formula:

$$a_n = a + (n - 1)d$$

Example:

In the sequence **2, 5, 8, 11, ...,** the common difference d is **3**.

Geometric Sequence

- **A geometric sequence** (progression) is a sequence of numbers where each term after the first is found by multiplying the previous term by a fixed, non-zero number called the "common ratio," denoted by *r*. The general form of a geometric sequence is:

$\boldsymbol{a, ar, ar^2, ar^3, \ldots,}$ here, *a* is the first term of the sequence.

The n^{th} term of a geometric sequence can be calculated using the formula:

$$a_n = a \cdot r^{(n-1)}$$

Example:

In the sequence **3, 6, 12, 24, ...** the common ratio *r* is **2**.

Introduction to Set Theory

Sets are fundamental objects in mathematics, defined as collections of distinct *elements*.

The elements that make up a set can be anything: *numbers, letters, other sets, and so on.*

Sets are typically denoted using *curly braces*, with elements separated by commas.

Examples: {a,b,c} {☼, 5, ♣} {1, 2, 3, 5, 7}

Sequences vs Sets

Property	Sets	Sequences
Definition	A collection of distinct objects.	An ordered list of elements, possibly with repetition.
Ordering	Order of elements is not significant.	Order of elements is significant.
Repetition	No repetition of elements is allowed.	Repetition of elements is allowed.
Notation	Typically denoted using curly braces {}{}.	Represented using parentheses ()(), angle brackets <><>, or listed.
Membership	An element either belongs or does not belong.	An element can occur multiple times at different positions.
Indexing	Elements are not indexed.	Elements are indexed; there is a first element, a second element, etc.
Cardinality/Length	Cardinality is the number of distinct elements.	Length is the total number of elements, including repetitions.

Some well-known and named sets

"N" is the natural numbers {0, 1, 2, 3, 4, 5, …}

"Z" is the set of integers {…-2,-1,0,1,2,…}

"Q" is the set of rational numbers

"R" is the set of real numbers

Empty Set. Cardinality

There are some special sets, such as the **empty set** $\emptyset$ or $\{\}$, which has no elements, and **universal sets (U)**, which contain all elements under consideration.

The **cardinality of a set** is the number of elements in the set. It is denoted as $|A|$ for a set A.

NB: If the *empty sets* are included on purpose as the elements of the sets, they are counted as separate elements and are included in set cardinality.

LECTURE 2 ACTIVITY:
Solve the following problems

$$A = \{alpha, beta, gamma\}$$

$$B = \{-5, 0, 5, 10\}$$

$$C = \{\{a, \{b\}\}, \{c, d\}\}$$

$$D = \{\{\}, \{\}, 10, 11\}$$

$$E = \{\}$$

1. What is |A|, |D|?
   |A|=3, |D|=3
2. What is |B|+|C|
   6
3. What is |D|+|E|-|A|?
   0

Subsets

A **set** *A* is considered a **(weak) subset of set B** if every element of set A is also an element of set B.

The notation used to represent this relationship is **$A \subseteq B$**, which can be read as "A is a subset of B".

This is analogous to the relation "less or equal."

Examples:

- If $A=\{1,2\}$ and $B=\{1,2,3,4\}$, then $A \subseteq B$, because every element of set A (which are 1 and 2) is also an element of set B.
- The empty set $\emptyset$ is a subset of every set because it does not contain any elements that could possibly be not included in any other set.

Proper Subset

If A is a subset of B, and A is not equal to B (meaning some element in B is not in A), then A is called a **proper subset of B**.

The notation used for a proper subset is **A⊂B**.

This is analogous to the relation “less”.

Examples:

- If A={1,2} and *B*={1,2,3}, then A⊂B because A is a subset of B, and there exists an element in B (which is 3) that is not in A.
- Remember, by this definition, every set is considered a subset of itself (A⊆A) since all elements of *A* are indeed in A. However, a set is not a proper subset of itself.

Equal sets

A set A is **equivalent** to a set B (abbreviated as **A ≡ B**) if the following conditions hold

1) Every element of A is an element of B

2) Every element of B is an element of A

Examples:

$\{1,2,3\} \equiv \{1,2,3\}$ as both conditions above hold

{♣ ,☼, 5} ≡ {☼, 5, ♣}