What Is the Science of Programming

- The Science of Programming is about **program verification**. Program verification aims to get *reliable* programs by discerning properties about programs. In other words, it is about *reasoning* about correctness of programs as we write them: for each piece of code we write, we make sure that it gives us what we want.

- For this class, we'll look at a simple programming language.
  - The *syntax* will be simple because that's not the important part. What's important is formally (mathematically, logically) specifying the *semantics* of programs and connecting them to the semantics of logical statements.
  - As an aside: syntax means something related to grammar and semantics means something related to meaning.

- Test/debug vs Verification.
  - Test + debug can be extremely time consuming.
  - Testing cannot show correctness unless we have cases covering all possible situations; this might be a very large set. So, we reason about our programs to identify a practical number of test cases that should represent all the possible test cases.
  - On the other hand, we can overlook cases while verifying programs (because the rules we use can be too strict). We need testing as a reality check to show that our reasoning is sound.

1. We have a small program: "add integer x to z , but only if x is positive" and our specification is "$z \geq c$ before the program, and $z > c$ after it." How to decide test cases?

   - We can write in this program in pseudo-code or any programming language easily:

```
# z >= c
if x > 0:
    z += x
# z > c
```

   - We can have infinite numbers of test cases since there are infinite numbers of integers.
   - We can reason about how the statements and properties interact: take $x > 0$ (and its negation $x \leq 0$) and break up $\leq$ into separate $<$ and $=$ cases ($x > 0, x = 0$), to get $x < 0$, $x = 0$, and $x > 0$ as the general set of cases. If we think $x = -1$ and $x = 1$ are good enough generalizations of $x < 0$ and $x > 0$, then we're done: Our test cases are $x =- 1, x = 0, x = 1$.

   - Of course, we can think of $x = 1$ and $x > 1$ as two different cases (similarly $x = -1$ and $x < -1$); we can use $x = 2$ and $x = -2$ to test the cases $x > 1$ and $x < -1$ and we end up with five test cases: $x = -2$, $x = -1$, $x = 0$, $x = 1$, $x = 2$. If we keep breaking the cases, we will get an infinite number of cases; a big part of testing is figuring out when to stop doing all this.

   - As an aside, in this class we will use rules to prove that $\{z \geq c\}$ **if** $x > 0$ **then** $z = z + x$ **fi** $\{z > c\}$ is always valid so we don't need to test this piece of code at all.

Review of Propositional Logic

- **Logic** is the study of formal or valid reasoning. A **proposition** is a declarative sentence that is either true or false.
    - A paradox is not a proposition. For example, "this is sentence is wrong" is neither true nor false.
- **Propositional logic** is logic over **proposition variables** (usually expressed with letters p, q, r ...), which are just variables that can have the values true or false.
    - In computer science terms, propositional logic is the logic used for Boolean expressions: $True$ and $False$ are Boolean constants.
- In propositional logic we study the **propositional operators**: and (∧), or (∨), not (¬), implication (→), and biconditional (↔). Instead of discussing all the operators one by one, let's answer the following questions together.

2. Which of the following are propositions?

a.	IIT was founded in 1940.	Yes, and it is $False$.
b.	Are you still playing Pokemon Go?	No, it is a question.
c.	Please log on to myIIT with your hawk credentials.	No, it is a command.

3. With the propositions

    p: It is below freezing. q: It is snowing.

    Represent the following propositions using p, q, and logical connectives.

    (a) It is below freezing and snowing. $p \land q$
    (b) It is below freezing but not snowing. $p \land \neg q$
    (c) It is not below freezing, and it is not snowing. $\neg p \land \neg q$
    (d) It is either snowing or below freezing (or both). $p \lor q$

4. Translate each of the following to either $p \to q$ or $q \to p$:

    a. if p then q $p \to q$
    b. p is sufficient for q $p \to q$
    c. p only if q $p \to q$
    d. p is necessary for q $q \to p$
    e. p if q $q \to p$

    - The following propositions all means $p \to q$.
        - "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 $\neg p$"

- $p \leftrightarrow q$ is the **biconditional** of p and q, it means $p \to q$ and $q \to p$ at the same time; in other words, $p \leftrightarrow q$ is $True$ when p and q are both $True$ or both $False$. But it doesn't mean p and q are "the same".