

## Theory Jigsaw - possible entries

	Universality	Computability	Intractability	Programming Languages
Big Idea	General purpose computer	Some problems are non-computable	Some problems take too long	In theory all equivalent In practice, different features matter
	Church-Turing Thesis			
Vocabulary	thesis	undecidable	search problem	type checking
	model of computation	unsolvable	P (efficient)	compiler vs interpreter
		paradox	NP	object oriented programming
		proof by contradiction	NP-complete	immutable
Practical Consequence	Same machine can be used for many jobs	No guaranteed virus detection	Take advantage of intractability e.g., RSA encryption	No single perfect language (depends on the job) (so learn more than one)
	Different machines can be used for same job			
People	Alan Turing Turing Machine	Alan Turing Halting Problem undecidable	Dick Karp 3SAT reduces to a set of problems	Alan Kay promoted OOP, conceived notebook computer (1970s)
	Alonzo Church lambda calculus	David Hilbert all math solvable (wrong)	Stephen Cook Set of problems reduces to 3SAT	Charles Simonyi encode type in variable name, MicroSoft Word code
	Ada Lovelace idea of general purpose computers	Kurt Godel Incompleteness Theorem		Bjarne Stroustrup C++ James Gosling Java

Church-Turing Thesis: Turing machines can do anything that can be described by a purely mechanical process. (i.e., TM as powerful as any other machine.)

Extended Church-Turing Thesis: P is the set of search problems solvable in poly-time in this universe. (i.e., TM as efficient as any other machine.)