1. List five reasons that a methodology and notation are needed for describing the semantics of programming languages.

* Programmers need to know what statements mean
* Compiler writers must know exactly what language constructs do
* Correctness proofs would be possible
* Compiler generators would be possible
* Designers could detect ambiguities and inconsistencies

1. Define operational semantics.
   1. Describe the meaning of a program by executing its statements on a machine, either simulated or actual. The change in the state of the machine (memory, registers, etc.) defines the meaning of the statement
2. Why can machine languages not be used to define statements in operational semantics?

* Machine languages are not used to define statement in operational semantics because it is too low-level to easily understand and it is difficult to read and write.

1. Describe the two levels of uses of operational semantics.

Natural operational semantics (at the highest level, we are interested in the final result of the execution of a complete program)

- Structural operational semantics (at the lowest level, we are interested in determining the precise meaning of a program through an examination of the complete sequence of state changes that occur when the program is executed)

1. Evaluate operational semantics.
   1. Good if used informally (language manuals, etc.)
   2. Extremely complex if used formally (e.g., Vienna Definition Language or VDL, which was used for describing semantics of PL/I.
2. Define denotational semantics.

* Based on recursive function theory
* The most abstract semantics description method
* The most widely known formal method for describing the meaning of programs.
* Originally developed by Scott and Strachey (1970)

1. What two things must be defined for each language entity in order to construct a denotational description of the language?

* Define mathematical object for each language entity.
* Define a function that maps instances of the language entities onto instances of the corresponding mathematical objects.

1. In denotational semantics, what are the syntactic and semantic domains?

The mapping functions of a denotational semantics programming language specification, like all functions in mathematics, have a domain and a range.

The domain is the collection of values that are legitimate parameters to the function; the range is the collection of objects to which the parameters are mapped.

In denotational semantics, the domain is called the **syntactic domain**, because it is syntactic structures that are mapped.

The range is called the **semantic domain**.

1. In what fundamental way do operational semantics and denotational semantics differ??R

In operational semantics, the state changes are defined by coded algorithms

In denotational semantics, the state changes are defined by rigorous mathematical functions

1. Evaluate denotational semantics.

* Can be used to prove the correctness of programs
* Provides a rigorous way to think about programs
* Can be an aid to language design
* Has been used in compiler generation systems
* Because of its complexity, it is of little use to language users

1. Define axiomatic semantics.

* Based on formal logic (predicate calculus)
* Original purpose: formal program verification
* Axioms or inference rules are defined for each statement type in the language (to allow transformations of logic expressions into more formal logic expressions)
* The logic expressions are called *assertions*

1. Which part of an inference rule is the antecedent?

The first part of a logic rule or assertion is called the antecedent and the second part is called the consequent. If A then B.

1. What is an axiom?

An axiom is a logic rule or assertion that is assumed to be true (fact). Therefore, an axiom is an inference rule without an antecedent.

1. On what branch of mathematics is axiomatic semantics based?

* Predicate calculus

1. On what branch of mathematics is denotational semantics based?

* Function

1. Explain what the preconditions and postconditions of a given statement mean in axiomatic semantics.

An assertion before a statement (a *precondition*) states the relationships and constraints among variables that are true at that point in execution

An assertion following a statement is a *postcondition. The post codition for the entire program is the desired result.*

1. Describe the approach of using axiomatic semantics to prove the correctness of a given program.

The postcondition for the entire program is the desired result

* 1. Work back through the program to the first statement. If the precondition on the first statement is the same as the program specification, the program is correct.

1. Evaluate axiomatic semantics.

* Developing axioms or inference rules for all of the statements in a language is difficult
* It is a good tool for correctness proofs, and an excellent framework for reasoning about programs, but it is not as useful for language users and compiler writers
* Its usefulness in describing the meaning of a programming language is limited for language users or compiler writers

1. Which semantics approach is most widely known?

* Denotational semantics