Answer Key to Assessment 8

Solution to Question #1

Verification focuses on the correctness of the translation. For a compiler or translation software, this includes:

a) detecting any lexical, syntactic, or semantic errors in the input language;

b) generating correct syntax per the output language; and

c) generating an output language program that has equivalent semantics to that of the input language program.

Verification consists of verifying:

1) individually each part of the compiler/translator – lexor, syntax analyzer, semantics analyzer, code generators, optimizers;
2) integration of the parts work correctly together;
3) correct operation in the various required target configurations for various required operating systems, and machines (architectures); and
4) the compiler and its components interface correctly with related system and application software.

Verification can include various techniques:

1. formal grammars, proofs using type systems, reuse of prior verified components, and generators;
2. design and code walkthroughs;
3. peer reviews;
4. tests using formal test suites as well as custom application specific test suites; and
5. pilot operations.

Solution to Question #2

Validation includes correctness (verification) but also is much broader in that it focuses on demonstrating the satisfaction of the goals/objectives and the requirements for the compiler or translation software, in order to address the expectations of all relevant stakeholders. These can include:

1. performance characteristics of both the development of the compiler, as well as the operation of the compiler in customer environments and on required networks;
2. maintainability of the compiler;
3. other design criteria, including reliability, availability, complexity, usability, portability, and extensibility;
4. efficiency of the compiler in terms of memory requirements and timing; also efficiency of the compiled program itself;
5. scalability in terms of number of concurrent users and the size of programs to be compiled;
6. development schedule; and
7. development cost and life cycle costs.

Validation techniques include the same techniques used for verification but oriented to the goals/objectives and user requirements. Moreover, when used for validation, these techniques have the participation of the relevant stakeholders, and they may even be conduced by stakeholders, e.g. the customer or users, or, even by an independent third party.

Students should appreciate the challenges involved in verification and validation, and that they mitigate technical, management, and business risks that confront the development of the compiler, its deployment, and operation.