ILLINOIS LICENSURE TESTING SYSTEM
FRAMEWORK AND OBJECTIVES
FIELD 838: COMPUTER SCIENCE

I. COMPUTER SYSTEMS

A. Structure of Computer Systems and Computer Networks

1. Identify the basic features of computer systems.

1.1 Identify and analyze the basic functions of computer systems (e.g., calculating, storage of information, communication, control).

1.2 Identify and analyze the basic components of computer systems (e.g., CPU, memory, input and output devices).

1.3 Identify the basic functions and characteristics of stand-alone microcomputer systems (e.g., with respect to external storage, memory size, file sharing, speed).

1.4 Identify and apply criteria for evaluating computer systems (e.g., storage capacity, execution speed, user characteristics).

2. Identify the general architectural characteristics of computers.

2.1 Identify and analyze the components of processing devices (e.g., control unit, arithmetic/logic unit) and their functions.

2.2 Identify types and analyze characteristics of storage devices (e.g., primary and secondary memory, RAM, ROM, hard disk, CD-ROM).

3. Identify types and characteristics of peripheral devices.

3.1 Identify types of input devices and analyze their characteristics (e.g., joystick, light pen, graphics tablet, keyboard, optical scanner, mouse, voice recognition unit).

3.2 Identify types of output devices and analyze their characteristics (e.g., video display terminals, printers, plotters, voice synthesizers).

3.3 Define and apply terms related to input and output devices (e.g., pixel, dot matrix, buffer, on-line, resolution).

3.4 Identify types and characteristics of telecommunications devices (e.g., modems, controllers, switches, PBX).

42. Understand the general principles of network connectivity.

42.1 Analyze the historical development of computer networks, including the Internet.

42.2 Identify and analyze the characteristics of a computer network (e.g., communication protocols, topologies, transmission media).

42.3 Identify network components and analyze their characteristics (e.g., workstations, file servers, bridges).

42.4 Identify and apply knowledge of the Internet (e.g., Email, newsgroups, World Wide Web).

B. Software

4. Identify types of programming software and software applications.

4.1 Identify and analyze types of software packages and their applications (e.g., word processing, spreadsheet, database management, fourth-generation languages, object-oriented languages, integrated packages, telecommunications packages).

4.2 Select appropriate software for a given situation.

4.3 Identify and analyze the functions of compilers, interpreters, assemblers, linkers, editors, and loaders.

4.4 Identify and analyze the functions of macros.

5. Identify criteria for evaluating software.

5.1 Identify and apply criteria for evaluating educational software.

5.2 Identify and apply criteria for evaluating applications software (e.g., spreadsheet, word processing).

5.3 Identify and apply criteria for evaluating telecommunications software (e.g., Internet access).

6. Understand the functions and characteristics of operating systems.

6.1 Identify and analyze various functions of an operating system (e.g., maintaining file systems, managing processor and memory, program execution, job control language).

6.2 Identify types and analyze characteristics of operating systems utilities (e.g., library commands, utility programs).

II. PROGRAMMING THEORY AND METHODS

A. Program Specifications, Design, and Coding

7. Understand the function of flowcharts and pseudocode in program preparation.

- 7.1 Identify and analyze the functions of flowcharts and pseudocode in program preparation.
- 7.2 Interpret computer flowcharts.
- 7.3 Interpret pseudocode.

8. Apply modularization in program design.

- 8.1 Identify the function of the module in program design.
- 8.2 Apply methods for decomposing programs into modules.
- 8.3 Apply methods for decomposing modules into submodules.
- 8.4 Analyze modularization strategy in a given programming situation.

9. Analyze design strategies in program development.

- 9.1 Identify and analyze the steps in a programming process.
- 9.2 Identify the purpose and function of design strategies (e.g., top-down, bottom-up, heuristic).
- 9.3 Distinguish between top-down and bottom-up designs.
- 9.4 Analyze the advantages of different design strategies.

10. Apply the principles of coding in program development.

- 10.1 Analyze the structure of program code (e.g., loops, subroutines).
- 10.2 Identify and apply techniques for developing code.

B. Testing, Debugging, and Documentation

11. Understand the principles of testing computer programs.

- 11.1 Identify the function of program testing.
- 11.2 Identify and apply strategies for designing test plans including interactive and noninteractive programs (e.g., testing interactions between prototype modules).
- 11.3 Sequence the steps in running a program test.
- 11.4 Identify and apply criteria for evaluating test plans.

12. Apply debugging procedures.

- 12.1 Identify and analyze common programming errors.
- 12.2 Apply procedures for locating program errors.
- 12.3 Identify and apply strategies for debugging programs.
- 12.4 Identify program errors in program listings and printed output.

13. Identify principles of program documentation.

- 13.1 Identify reasons for documenting computer programs.
- 13.2 Identify types and analyze characteristics of program documents (e.g., functional specifications, record layout, report layout, screen layout).
- 13.3 Identify types and analyze characteristics of internal documentation (e.g., using mnemonic identifiers, comment statements).
- 13.4 Identify types and analyze characteristics of external documentation (e.g., tutorials, manuals, operational procedures).

III. COMPUTER LANGUAGES, DATA CONCEPTS, AND ALGORITHMS

A. Programming Operations and Languages

14. Analyze declarations and data types common to high-level languages using a pseudolanguage.

14.1 Identify the function and analyze characteristics of variable declarations and data types.

14.2 Recognize the function and analyze characteristics of identifiers.

14.3 Interpret code for correct use of global and local identifiers.

15. Understand constants and variables common to high-level languages using a pseudolanguage.

15.1 Identify types and analyze characteristics of constants and variables.

15.2 Interpret code for appropriate use of data as represented by constants and variables.

16. Understand statements and operators common to high-level languages using a pseudolanguage.

16.1 Identify purposes of statements and operators.

16.2 Identify types and analyze characteristics of operators (e.g., arithmetic, Boolean, relational).

16.3 Identify and analyze the purpose and function of operator precedence.

16.4 Identify the purpose and function of assignment statements.

17. Understand control structures common to high-level languages using a pseudolanguage.

17.1 Identify and analyze characteristics of sequential execution structures (e.g., compound statements).

17.2 Identify and analyze characteristics of conditional execution structures with or without branching.

17.3 Identify and analyze characteristics of repetitive execution structures.

17.4 Interpret code for correct use of sequential, conditional, or repetitive execution control structures.

18. Identify input and output procedures common to high-level languages using a pseudolanguage.

18.1 Identify the function and analyze characteristics of terminal input and output procedures.

18.2 Identify the function and analyze characteristics of file input and output procedures.

19. Understand subprograms, procedures, and functions common to high-level languages using a pseudolanguage.

19.1 Identify and analyze the purpose and function of parameters.

19.2 Interpret code for the correct use of parameter passing techniques.

20. Apply methods of program annotation common to high-level languages using a pseudolanguage.

20.1 Identify and analyze the role and function of comments.

20.2 Identify and analyze the role and function of formatting and indentation.

20.3 Apply program annotation (e.g., comments, format and indentation) in a given situation.

43. Identify and analyze features and uses of common programming languages.

43.1 Identify and analyze programming features and applications of BASIC.

43.2 Identify and analyze programming features and applications of LOGO.

43.3 Identify and analyze programming features and applications of PASCAL.

43.4 Identify and analyze programming features and applications of C.

B. Data Types and Structures

27. Understand the internal representation of data.

27.1 Identify types and functions of number systems (e.g., binary, hexadecimal, octal).

27.2 Apply number systems skills (e.g., conversion between systems).

27.3 Identify and analyze characteristics of codes (e.g., ASCII, EBCDIC).

28. Identify the functions and uses of data structures.

28.1 Identify and analyze the functions of data structures (e.g., arrays, strings, linked lists, stacks, queues).

28.2 Identify and analyze the uses of data structures.

29. Understand representation of data structures.

29.1 Apply procedures for representation of data structures.

29.2 Identify and analyze characteristics of representation of data structures.

C. Algorithms

30. Use algorithms to manipulate data.

- 30.1 Identify and apply techniques for manipulating data structures using string processing (e.g., concatenation, substring extraction, matching).
- 30.2 Identify and analyze algorithmic operations commonly performed on data structures (e.g., insertions and deletion of data in linear structures and trees, tree traversals).
- 30.3 Apply procedures for manipulating data structures.

31. Identify characteristics and functions of algorithms.

- 31.1 Identify and analyze characteristics and functions of searching algorithms.
- 31.2 Identify and analyze characteristics and functions of sorting algorithms.

IV. COMPUTER APPLICATIONS AND ASSESSMENT A. Computers and Society

32. Understand the historical development of computers.

- 32.1 Identify major persons and events in the development of computers (e.g., Pascal, Babbage, Lovelace, UNIVAC, ENIAC).
- 32.2 Identify the generations of computer development and analyze their characteristics (e.g., development of transistors during the second generation).
- 32.3 Distinguish between analog and digital computers.
- 32.4 Identify and analyze significant technological aspects of the development of personal computers (e.g., microprocessor chip).

33. Analyze the impact of computers on society.

33.1 Analyze the growth of the computer industry.

33.2 Analyze the impact of computers on communication and transportation (e.g., satellite communication, automobiles, credit networks).

33.3 Analyze the impact of computers on entertainment and work life.

33.4 Analyze the impact of computers on the job market.

34. Analyze legal and ethical issues related to computers.

34.1 Analyze legal and ethical issues of professional conduct in the computer industry.

34.2 Analyze legal issues related to copyright in the computer field.

34.3 Analyze ethical issues concerning the use of computers in contemporary society (e.g., confidentiality legislation, illegal access, equity of access issues).

35. Analyze computer applications in business and industry.

35.1 Identify and analyze the uses of computers in data processing (e.g., data analysis, database management, text processing).

35.2 Identify and analyze the uses of computers in accounting and financial modeling.

35.3 Identify and analyze computer applications in industry and manufacturing (e.g., robotics, automated production control, CAD/CAM techniques).

35.4 Analyze the uses of telecommunication.

36. Identify computer applications in science, health, and the arts.

36.1 Identify and analyze the applications of computers using simulations and modeling.

36.2 Identify areas of research in artificial intelligence (e.g., heuristic programming, expert systems, robotics, cybernetics).

36.3 Identify and analyze computer applications in the health field (e.g., expert systems, computer-enhanced diagnosis).

37. Analyze career opportunities involving computer science.

37.1 Identify types of careers involving computer science.

37.2 Identify the requirements of careers in computer science.

37.3 Analyze the impact of social and technological change on computer professions.

B. Computers and Education

38. Understand issues in computer education.

38.1 Identify basic issues of computer literacy (e.g., computer use vs. programming ability).

38.2 Identify the goals and objectives of various computer curricula (e.g., data processing, programming, literacy).

39. Analyze instructional uses of computers.

39.1 Analyze the uses of computer-assisted instruction.

39.2 Analyze types and characteristics of computer-assisted instructional models (e.g., drill and practice, simulation).

39.3 Analyze the uses of computer-managed instruction (e.g., testing, scoring, record keeping).

39.4 Analyze ways to use computers in specific subject areas (e.g., science, English, art).

40. Analyze the uses of computers in special education.

40.1 Identify types and characteristics of students with special educational needs (e.g., learning disorder, physical disability).

40.2 Identify types and analyze characteristics of computerized devices used in special education (e.g., for communication).

40.3 Identify and apply instructional activities involving computers for students with special educational needs.

40.4 Analyze the benefits of computers in special education.

41. Identify the professional responsibilities of computer science teachers.

41.1 Identify and apply ways to evaluate, select, and develop instructional materials which involve using computers.

41.2 Identify and apply ways to assist the school in evaluating, selecting, and acquiring computer equipment.

41.3 Identify criteria for evaluating the computer science program.

41.4 Identify the functions and characteristics of professional organizations and publications involved in computer science (e.g., Association of Computing Machinery).

C. Assessment

44. Understand principles of measurement and evaluation as applied to instruction, assessment, and program evaluation.

- 44.1 Apply major terms and concepts associated with educational measurement and evaluation.
- 44.2 Apply statistical principles needed for the development, selection, and interpretation of educational tests.

45. Understand how to select and develop fair, effective, and appropriate educational assessment instruments.

- 45.1 Apply criteria and procedures for the selection of educational tests, the creation of test frameworks and test objectives, and the development of individual assessments.
- 45.2 Analyze the uses and limitations of a variety of assessment instruments and processes.
- 45.3 Analyze the relationship of assessment to instruction and the alignment of assessment instruments with the curriculum.
- 45.4 Apply principles of nondiscriminatory test construction.

46. Understand how to administer, score, and interpret a variety of educational assessment instruments.

- 46.1 Apply accepted test administration procedures, including the preparation of testing accommodations for special populations.
- 46.2 Apply accepted test scoring procedures and interpret commonly reported scores.

47. Understand how to use assessment data and information to promote student achievement as it relates to educational planning and school improvement.

47.1 Analyze the use of assessment data and information to identify individual student strengths and weaknesses.

47.2 Identify the role of assessment data and information in school building and district program evaluation and instructional planning.

47.3 Apply techniques for communicating appropriate assessment data and information to parents/guardians, staff, and the community.