

Nebraska Department of Education
Classification Specification
Established: 9/02 Revised: 6/0714

Job Code: EDB101/EDR101/EDO101
Paygrade: 46

IT Applications Developer

Purpose of Class: Performs programming for computer applications; participates in design and analysis of computer applications; participates with other analysts in maintaining and enhancing existing computer applications; incumbents primarily assist higher-level personnel in the series; all work is reviewed by others.

Distinguishing Characteristics

Level: First in a series of three

Work Direction Received: Works under general supervision

Direction of Others: None

Scope/nature of Discretion: Limited discretion; performs duties with some/moderate opportunity to exercise independence within well-defined boundaries

Examples of Duties

General Duties

- Consults with clients on application issues
- Uses application development tools and utilities as appropriate

Accomplishes the following tasks and develops quality products of typical scope and complexity with limited assistance:

System Design and Implementation

- Functional design
- Technical design
- System specifications
- Develop implementation plan
- Training
- Support material
- On-going support

Independently accomplishes the following tasks of typical complexity, with minimal assistance, in a competent manner:

Programming and testing

- Develops, debugs, assembles/compiles code
- Tests programs
- Documents programs

Communicates with peers, clients and management

Examples of Duties (continued)

Performs the following technical tasks with considerable assistance from others:

Systems Analysis

- Business problem definition
- Information analysis and organization
- Information collection
- Economic evaluation
- Solution development and evaluation

Logical Database Design, in conjunction with IT Data/Database Analysts

- Data analysis
- Logical views
- Data management
- Data structure
- Data definition

Physical Database Design, in conjunction with IT Data/Database Analysts

- Physical design
- Access methods
- Sub-system facilities
- Relational databases

Programming and testing

- Develops program code, assembles/compiles code, and debugs code
- Performs alpha/unit testing and participates in system program testing
- Prepares program documentation

Maintains regular and reliable attendance.

Minimum Qualifications

Associate's degree in a related area such as computer science, computer engineering, or management information systems. Any equivalent education and/or work experience may be substituted in order to meet the minimum qualifications of the position.

Optional: Some positions may require coursework, training or work experience in certain programming languages. Lesser degrees may be considered with increasing years of experience as follows:

Degree	Years of Programming Work Experience
Associate's degree in an unrelated area	2 years
High school diploma	4 years

Formatted: Left

Formatted: Left

Formatted: Left

Knowledge and Abilities

Applicants will be screened through written, oral, performance and/or other evaluations for the following:

Knowledge

- Basic math
- Software development principles and methods
- Design techniques, principles, tools and instruments
- Principles, methods and procedures for designing, developing, optimizing and integrating new/reusable systems components
- Software system testing procedures, programming and documentation

Knowledge (continued)

- Database principles and methodologies
- Systems engineering concepts and factors

Abilities

- Use basic math to solve problems
- Write various computer programs
- Apply programming languages
- Read, listen, and understand written/oral communication from others
- Communicate, both orally and in writing, information/ideas to others
- Apply general rules to specific problems to arrive at logical conclusions; decide if an answer makes sense; identify discrepancies
- Order information; organize different pieces of information into a meaningful pattern
- Concentrate without distraction while performing tasks over a period of time
- Organize, plan and prioritize work
- Test, install, implement, document, and maintain software
- Maintain source code; modify and upgrade code as necessary
- Active learning/working with new material/information to grasp its implications
- Write, debug and maintain code
- Integrate hardware/software components
- Determine output media/formats