DRAFT CHANGE OF APPROACH

SOFTWARE DEVELOPER
(Time-Based)

APPENDIX A

O*NET CODE 15-1131.00

This training outline is a minimum standard for Work Processes and Related Instruction. Changes in technology and regulations may result in the need for additional on-the-job or classroom training.

Potential Job Titles: Computer Programmer, Software Coder, Web Programmer.

WORK PROCESSES

Approximate Hours

A. Workplace Basics **140**

1. Describe workplace structure.
2. Describe workplace policies and procedures; general and
3. Information Technology (IT) -related.
4. Demonstrate an understanding of general ideas regarding workplace ethics, interpersonal communication, and personal safety.
5. Demonstrate efficient basic task/time management, status reporting, work order updates, team participation.
6. Demonstrate ability to communicate technical ideas/concepts when assisting users unfamiliar with IT jargon.

B. Defining Projects **50**

1. Identify and record customer/user/stakeholder requirements.
2. Compile set of potential actions users may take in a system.
3. Compose requirement specifications (if applicable).

C. Designing Software **60**

1. Identify and record requirements.
2. Recognize computer hardware and demonstrate knowledge of same.

3. Identify potential security threats and other vulnerabilities which may arise.

D. Writing Software Code **950**

1. Set up programming environment to be used with selected software, such as: C++, Java, Python, etc.
2. Input and store data; name and define four main data types:
   a. string
   b. integer
   c. floating-point number
   d. Boolean
3. Convert data from one type to another.
4. Recognize and correct errors.
5. Perform operations, such as: arithmetic, assign values, compare values, find truth (logic), set order.
6. Make lists: write, change, fix, set, retrieve.
7. Construct branch choices.
8. Write loops, including breaks and skips.
9. Create and store custom functions.
10. Import functions (if applicable).
11. Write various types of sorting algorithms.
12. Import pre-defined functionality into programs.

E. Building Graphical Interfaces and Applications **400**

1. Write interface code to produce display messages, gather entries, check boxes, and add images to applications.
2. Develop applications; tests and deploy applications on different operating systems (if applicable).

F. Testing, Deployment, and Maintenance **400**

1. Develop new software.
2. Troubleshoot programs.
3. Train end users (if applicable)

Approximate Total Hours **2,000**

Apprenticeship work processes are applicable only to training curricula for apprentices in approved programs. Apprenticeship work processes have no impact on classification determinations under Article 8 or 9 of the Labor Law. For guidance regarding classification for purposes of Article 8 or 9 of the Labor Law, please refer to https://dol.ny.gov/public-work-and-prevailing-wage

SOFTWARE DEVELOPER

APPENDIX B

RELATED INSTRUCTION

Safety/Health/Environment

1. General Workplace Safety
2. Right-to-Know/ Safety Data Sheets (SDS)
3. Sexual Harassment Prevention Training – must comply with section 201-g of the Labor Law)

Computer and Network Components and Operations

1. Hardware
2. Peripherals
3. Software installation
4. Operating Systems, e.g., Microsoft, MacOS
5. Troubleshooting
6. Programming Languages, e.g., Java, C++, Python
7. Cybersecurity

Professional Development

1. Technical Support Communication
2. Time Management
3. Basic Project Management
4. Team and Supervisor Communication Skills
5. Customer Service Fundamentals

Other Courses as Necessary

A Minimum of 144 Hours of Related Instruction is Required for Each Apprentice for Each Year.

Appendix B topics are approved by New York State Education Department.