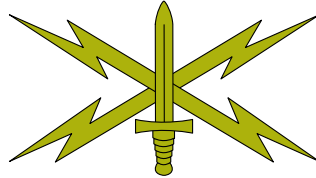


Job Qualification Record
for the work role
Cyber Capability Developer
Basic Level



Assigned to:	_____	Signature:	_____
Trainer Name:	_____	Signature:	_____
Date assigned:	_____		
Estimated completion date:	_____	Actual completion date:	_____
Evaluator:	_____		
Qualifier:	_____		

This JQR is required to be completed within _____ working days of receipt

JQR Version 1.1 as of 2021-04-07
JQR remote <https://gitlab.devforce.disa.mil/arcyber-work-role-jqs/ccd-basic-dev>
JQR commit 84a7113 in master branch at 2021-05-12 09:04:21 +0000

This document was compiled at 2021-05-12 05:41:14 by jqr_generator.py
Generator remote https://gitlab.devforce.disa.mil/arcyber-work-role-jqs/jqr_product.git
Generator commit 9b7d845 in master branch at 2021-05-12 05:05:24 -0400

CLASSIFICATION INSTRUCTIONS

This document's **first two pages are CUI** when separated from the rest of this document.
This document contains portion markings classified up to the **CUI** level.

Controlled by: U.S. Army Cyber Command
CUI Category: PERS (when completed)
Distribution/Dissemination Control: FEDCON
POC: usarmy.gordon.arcyber.mesg.g37-training-branch@mail.mil

(U) Module signature page

Module 1

Trainee Name:	_____	Signature:	_____
Site Lead Name :	_____	Signature:	_____
Start Date:	_____	Completion Date:	_____

Module 2 Administrative

Trainee Name:	_____	Signature:	_____
Site Lead Name :	_____	Signature:	_____
Start Date:	_____	Completion Date:	_____

Module 3 Cyberspace Capability Developer Basic Core

Trainee Name:	_____	Signature:	_____
Site Lead Name :	_____	Signature:	_____
Start Date:	_____	Completion Date:	_____

Module 4 Basic Cyberspace Capability Developer (CCD) Knowledge

Trainee Name:	_____	Signature:	_____
Site Lead Name :	_____	Signature:	_____
Start Date:	_____	Completion Date:	_____

Module 5 Army-specific operational training requirements and recommendations

Trainee Name:	_____	Signature:	_____
Site Lead Name :	_____	Signature:	_____
Start Date:	_____	Completion Date:	_____

Module 6 Record of Assessments and Formal Training Completion

Trainee Name:	_____	Signature:	_____
Site Lead Name :	_____	Signature:	_____
Start Date:	_____	Completion Date:	_____

(U) Release history

Ver	Date	OPR	Change
1.0	1 Jan 19	780th	Annual Update
1.1	7 Apr 21	780th	Annual Update

(U) Summary of Changes

Ver	Date	OPR	Change
1.0	1 Jan 19	780th	Annual Update
1.1	7 Apr 21	780th	<ul style="list-style-type: none">• Recert 3.6.1 : Added capability management process• Recert 4.5.4 : Added conditional variables• Recert 4.1.6 : Added pre and post incrementation• Recert 2.3.1, 2.4.1 : Added coding standards and style guides• Recert 4.1.19 : Added make and Makefiles• Recert 4.1.23 : Added additional memory management subtasks• Move to JSON-based JQR• Added generics to data structures• Removed sethostname• Added with statement• Removed env vars• Removed references to Windows data types and added references to stdint.h• Removed two of the three linked list line items• Removed Dijkstra's algorithm and added a priority queue• Added new standardized modules• Specify a Unix-based OS for file operations• Phrasing and grammar fixes• Added additional bitwise subtasks

Contents

1 (U) Work Role Information	5
1.1 (U) Contents	5
1.2 (U) Proficiency/Work Function	5
1.3 (U) Prerequisites and Requirements	5
1.4 (U) Work Description	5
1.5 (U) Skill Progression Roadmap	6
1.6 (U) Conditions	7
1.7 (U) Standards	7
2 (U) Module 2: Administrative	8
2.1 Accounts: Network and Database	8
2.2 Security Read-Ins	8
2.3 Required Python Coding Standard	8
2.4 Required C Coding Standard	8
3 (U) Module 3: Cyberspace Capability Developer Basic Core	9
3.1 Cyberspace Operations Fundamentals	9
3.2 Cyber Warfare Battalion Fundamentals	10
3.3 DCO Fundamentals	11
3.4 OCO Fundamentals	12
3.5 Mission Support Orientation	13
3.6 Mission Process	14
4 (U) Module 4: Basic Cyberspace Capability Developer (CCD) Knowledge	15
4.1 C Programming	15
4.2 Python Programming	22
4.3 Data Structures	24
4.4 Algorithms	28
4.5 Operating System Concepts	28
4.6 Secure Coding	30
4.7 Networking Fundamentals	30
4.8 Serialization	31
5 (U) Module 5: Army-specific operational training requirements and recommendations	32
5.1 There are no additional items in this module	32
6 (U) Module 6: Record of Assessments and Formal Training Completion	33
6.1 Basic Skill Level Exam	33
6.2 Basic Skill Level Validation Panel	33
7 (U) Appendix A: Glossary	34

1 (U) Work Role Information

1.1 (U) Contents

(CUI) This Job Qualification Record (JQR) is divided into 6 modules. The Administrative module contains administrative tasks that must be accomplished prior to JQR qualification. The Organizational Primer module contains the fundamental knowledge that every member of the Cyber Mission Force must know. The Basic Cyberspace Capability Developer (CCD) Knowledge module contains the fundamental knowledge needed to perform the mission, the performance tasks that must be accomplished in order to achieve JQR qualification, and the information on the tools, databases, and systems required to conduct the mission.

1.2 (U) Proficiency/Work Function

(CUI) The purpose of this JQR is to validate a CCD trainee's knowledge, skills and abilities required to perform at Basic Proficiency as a CCD. The scope of this JQR includes all assigned (civilian, military and contract) personnel within the development organization assigned to Cyber Solutions Development (CSD) organization, Cyber Warfare Battalion (CWB), Cyber Protection Brigade (CPB) or any other Army organization recognized as providing cyberspace capability development. Active mentorship is essential to a successful and efficient JQR process. The mentor/trainer (TRR) and mentee/trainee (MTE) should meet regularly to discuss the MTE's JQR progress, validate the MTE's work against JQR line items, and prepare the MTE for the proficiency exam.

1.3 (U) Prerequisites and Requirements

(U) JQR Prerequisites:

(U) An individual interested in becoming a CCD must have documented programming knowledge through one of the following methods:

- A degree from an accredited institution in Computer Science, Electrical Engineering, or a related field
- Graduation from the Tool Developer Qualification Course (TDQC) or its derivatives
- Documented work experience commensurate with the line items in the applicable CCD JQR

(U) All prospects that did not graduate from TDQC will complete an initial entry assessment to assess their level of programming abilities. The results of the initial entry assessment will be used in tailoring their mentorship requirements and as documented evidence of their ability to meet individual requirements outlined within this JQR, where appropriate.

(U) JQR Requirements:

(U) To qualify as a Basic CCD, an individual must:

- Successfully complete 100% of each section of this JQR.
- Complete the JQR within the time allotted.
- Successfully complete the current Basic Skill Level Exam (BSLE) in accordance with approved practices and policies.
- Receive Basic Certification recommendation from validation panel.

1.4 (U) Work Description

(U) A CCD is an innovative, agile, highly skilled practitioner that conducts vulnerability analysis, exploitation research and development, software development, software and user documentation, and implementation of software and hardware capabilities that operate in and through cyberspace. CCD's and their capabilities are foundational elements of cyberspace operations and serve as a force multiplier for the Cyber Mission Force and conventional maneuver forces.

Trainer comments:

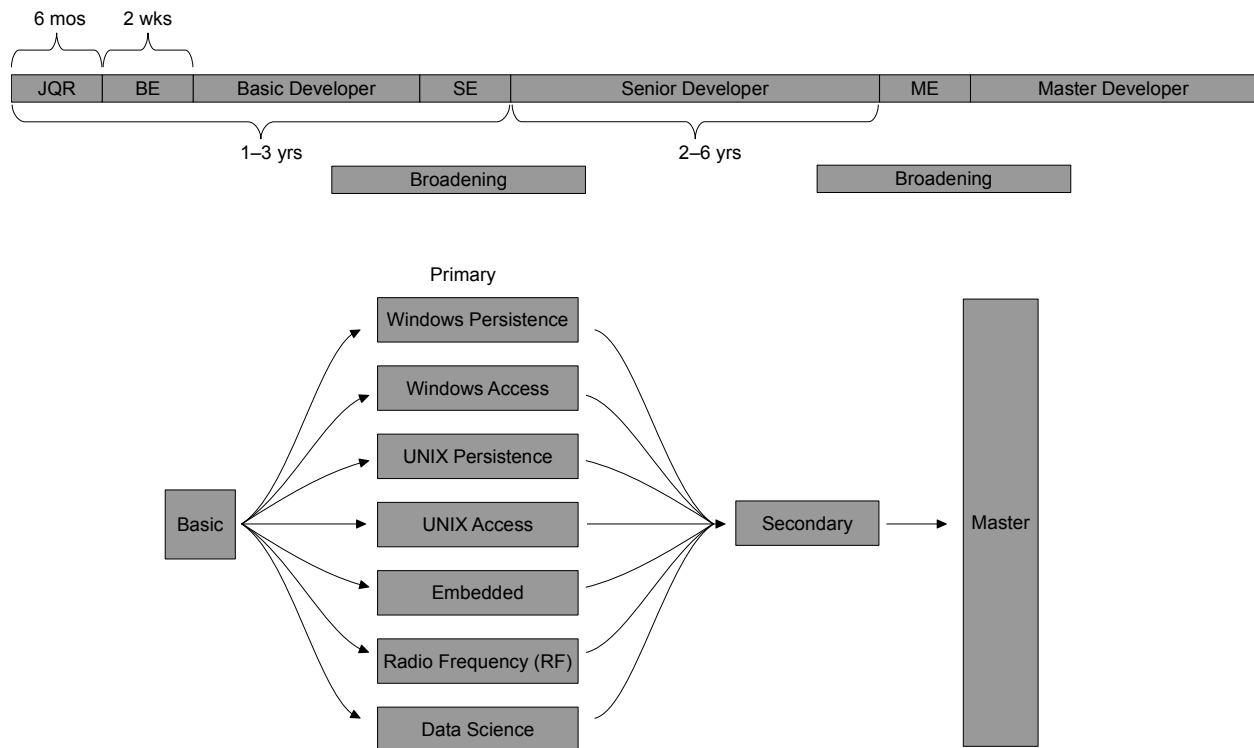


Figure 1: JQR Progression

1.5 (U) Skill Progression Roadmap

(U) CCDs advance in skill and ability through mentorship and experience as they progress and specialize in three skill levels: Basic, Senior, and Master (see Figure 1). They are required to stay current with technology and maintain their proficiency in the skill set in which they have certified. During transition between skill levels, CCDs have the option to participate in broadening assignments such as Computer Network Operations Development Program (CNODP), Training With Industry (TWI), U.S. Military Academy (USMA) Instructor, Researcher, Defense Digital Service (DDS) Advanced Education Program (AEP), Advanced Civil Schooling (ACS), and branch-specific Professional Military Education (PME). Additionally, prior to master skill level, a CCD must certify as a senior in a secondary specialty.

Trainer comments:

1.6 (U) Conditions

Condition	Description
A	The trainee must complete the line item without any references or resources.
B	The trainee must complete the line item with static personal, open-source, or institution resources in either written or digital form.
C	The trainee must complete the line item with the resources listed in condition B as well as a computer workstation with access to necessary tools and software.

1.7 (U) Standards

1.7.1 Standard A

0	1	2	3	4
Trainee has no familiarity with the term or task.	Trainee has some familiarity with the term or task but cannot accurately define the term.	Trainee is able to define the term but is unable to describe its use.	Trainee is able to accurately define the term or task and describe its use.	Trainee is able to accurately define the term or task, describe its use, and demonstrate advanced comprehension in the cognitive domain.
Needs retraining			Met Standard	Exceeds Standard

1.7.2 Standard B

0	1	2	3	4
Trainee writes code that does not compile or does not address the problem defined in the task.	Trainee writes code that compiles but does not address the task.	Trainee writes code that compiles and is related to the task but contains errors.	Trainee writes compiled code that completes the task.	Trainee writes compiled code that completes the task and includes additional functionality, improved efficiency, or accounts for edge cases.
Needs retraining			Met Standard	Exceeds Standard

1.7.3 Standard C

0	1	2	3	4
Trainee writes code that causes syntax errors or does not address the problem defined in the task.	Trainee writes code that executes without syntax errors but does not address the task.	Trainee writes code that executes without syntax errors and is related to the task but contains runtime errors.	Trainee writes code that executes without errors and completes the task.	Trainee writes code that executes without errors, completes the task and includes additional functionality, improved efficiency, or accounts for edge cases.
Needs retraining			Met Standard	Exceeds Standard

1.7.4 Standard D

0	1	2
Trainee does not have access to the related access/accounts or materials.	Trainee has the access to the account service or material and completed the task.	Trainee is able to use accounts/services effectively to complete a given task.

Trainer comments:

2 (U) Module 2: Administrative

Covers all initial requirements that must be met prior to completion of the JQR

2.1 Accounts: Network and Database

2.1.1 (CUI) Accounts <input type="checkbox"/> JWICS, NSANet, or equivalent <input type="checkbox"/> Public Key Infrastructure (PKI) if applicable <input type="checkbox"/> NIPRNet or equivalent <input type="checkbox"/> SIPRNet or equivalent <input type="checkbox"/> Gabriel Nimbus (NIPR/SIPR) or equivalent <input type="checkbox"/> Persistent Cyber Training Environment (PCTE) or equivalent <input type="checkbox"/> Rapid Cyber Development Network (RCDN) or equivalent	Trainee	Trainer	Date
Condition C – Standard D			

2.2 Security Read-Ins

2.2.1 Security Read-Ins <input type="checkbox"/> (CUI) Special Intelligence (SI) <input type="checkbox"/> (CUI) Talent Keyhole (TK) <input type="checkbox"/> As required by organization	Trainee	Trainer	Date
Condition C – Standard D			

2.3 Required Python Coding Standard

2.3.1 With references, resources, and a provided unit-level coding standard or style guide, identify coding standard violations in Python source code	Trainee	Trainer	Date
Condition B – Standard D			

2.4 Required C Coding Standard

2.4.1 With references, resources, and a provided unit-level coding standard or style guide, identify coding standard violations in C source code	Trainee	Trainer	Date
Condition C – Standard D			

Trainer comments:

3 (U) Module 3: Cyberspace Capability Developer Basic Core

3.1 Cyberspace Operations Fundamentals

Training Resources: • JP 3-12

3.1.1 Describe the mission of the following Cyber Mission Force (CMF) Elements: <input type="checkbox"/> Cyber National Mission Force (CNMF) <input type="checkbox"/> Cyber Combat Mission Force (CCMF) <input type="checkbox"/> Cyber Protection Force (CPF)	Trainee	Trainer	Date
	Condition A – Standard A		
3.1.2 Describe the mission of the following CMF team types: <input type="checkbox"/> Combat Mission Team (CMT) <input type="checkbox"/> National Mission Team (NMT) <input type="checkbox"/> Combat Support Team (CST) <input type="checkbox"/> National Support Team (NST) <input type="checkbox"/> Cyber Protection Team (CPT)	Trainee	Trainer	Date
	Condition A – Standard A		
3.1.3 Read and understand the following cyberspace operations doctrine: <input type="checkbox"/> JP 3-12 (Cyberspace Operations) <input type="checkbox"/> USCYBERCOM Operational Guidance 3-2 (DCO) <input type="checkbox"/> USCYBERCOM Cyber Warfare Publication (CWP 3-33.4)	Trainee	Trainer	Date
	Condition B – Standard A		
3.1.4 Describe the mission and responsibilities of key organizations: <input type="checkbox"/> U.S. Cyber Command (USCC) <input type="checkbox"/> Defense Threat Reduction Agency (DTRA) <input type="checkbox"/> Office of the Secretary of Defense <input type="checkbox"/> Director Operation Test and Evaluation (DOTE) <input type="checkbox"/> Federally Funded Research and Development Centers (FFRDC) <input type="checkbox"/> Central Intelligence Agency (CIA) <input type="checkbox"/> Open Source Center <input type="checkbox"/> Federal Bureau of Investigation (FBI) <input type="checkbox"/> Department of Homeland Security (DHS) <input type="checkbox"/> Defense Digital Service (DDS) <input type="checkbox"/> Army Research Lab (ARL) <input type="checkbox"/> National Security Agency (NSA) <input type="checkbox"/> National Air and Space Intelligence Center (NASIC) <input type="checkbox"/> Army Capability Manager - Cyber (ACM-Cyber)	Trainee	Trainer	Date
	Condition B – Standard A		

Trainer comments:

3.1.5 Demonstrate knowledge of U.S. Code and its application to the Intelligence Community (IC) and Cyberspace Operations (Title 10, 18, and 50)	Trainee	Trainer	Date
	Condition B – Standard A		
3.1.6 (CUI) Demonstrate the ability to properly use, store, and forward information using classification marking and handling caveats	Trainee	Trainer	Date
	Condition A – Standard A		
3.1.7 (CUI) Describe Title 10 and its importance	Trainee	Trainer	Date
	Condition B – Standard A		
3.1.8 (CUI) Describe Title 50 and its importance	Trainee	Trainer	Date
	Condition B – Standard A		
3.1.9 (CUI) Demonstrate working knowledge of the policies, laws, and the authorities to operate that govern capability development	Trainee	Trainer	Date
	Condition A – Standard A		

3.2 Cyber Warfare Battalion Fundamentals

3.2.1 Describe Expeditionary Cyberspace Team (ECT) concepts: <input type="checkbox"/> ECT mission <input type="checkbox"/> ECT task organization <input type="checkbox"/> Inherent capabilities and equipment <input type="checkbox"/> Deployment model <input type="checkbox"/> Combatant Command alignment	Trainee	Trainer	Date
	Condition B – Standard A		
3.2.2 Describe RF and EW development concepts, capabilities, and considerations: <input type="checkbox"/> Purpose <input type="checkbox"/> Common reachback process <input type="checkbox"/> Inherent capabilities and equipment <input type="checkbox"/> Supporting and supported organizations <input type="checkbox"/> Basic considerations for RF and EW development	Trainee	Trainer	Date
	Condition B – Standard A		

Trainer comments:

3.3 DCO Fundamentals

Training Resources: • CWP 3-33.4 • JP 3-12

3.3.1 Describe the overall mission of a Cyber Protection Team (CPT)	Trainee	Trainer	Date
Condition A – Standard A			
3.3.2 Describe the CPT Support Element and Mission Element and their capabilities:	Trainee	Trainer	Date
Condition B – Standard A			
3.3.3 Describe CPT mission types: <input type="checkbox"/> National CPT <input type="checkbox"/> DoDIN CPT <input type="checkbox"/> Combatant Command CPT <input type="checkbox"/> Service CPT	Trainee	Trainer	Date
Condition B – Standard A			
3.3.4 Describe DCO mission types and tactical tasks: <input type="checkbox"/> DCO-IDM <input type="checkbox"/> DCO-RA <input type="checkbox"/> Hunt <input type="checkbox"/> Clear <input type="checkbox"/> Enable Hardening <input type="checkbox"/> Assess	Trainee	Trainer	Date
Condition B – Standard A			
3.3.5 Describe the following DCO work roles: <input type="checkbox"/> Cyber Operations Planner <input type="checkbox"/> Network Analyst <input type="checkbox"/> Host Analyst <input type="checkbox"/> Analytic Support Officer <input type="checkbox"/> Data Engineer <input type="checkbox"/> Network Technician <input type="checkbox"/> All-Source Analyst	Trainee	Trainer	Date
Condition B – Standard A			

Trainer comments:

3.3.6 Describe the following DCO terms: <input type="checkbox"/> Mission Relevant Terrain-Cyber (MRT-C) <input type="checkbox"/> Key Terrain-Cyber (KT-C) <input type="checkbox"/> Active and Passive Defense <input type="checkbox"/> Critical Asset List <input type="checkbox"/> Defended Asset List <input type="checkbox"/> Prioritized Defended Asset List	Trainee	Trainer	Date
Condition B – Standard A			

3.4 OCO Fundamentals

3.4.1 Describe the following policy documents: <input type="checkbox"/> DOD Directive 5240.1R to include Questionable Intelligence Activities <input type="checkbox"/> DOD Directive 5240.1 <input type="checkbox"/> NSA/CSS Policy 11-1 <input type="checkbox"/> The FISA act of 1978 <input type="checkbox"/> (CUI) USSID CR1610	Trainee	Trainer	Date
Condition B – Standard A			
3.4.2 (CUI) Describe USSID SP0018 to include violations and reporting	Trainee	Trainer	Date
Condition B – Standard A			
3.4.3 (CUI) Describe the difference between conducting Cyberspace Intelligence, Surveillance, and Reconnaissance (C-ISR) and Cyberspace Surveillance and Reconnaissance (C-SR)	Trainee	Trainer	Date
Condition B – Standard A			
3.4.4 Describe the following OCO work roles: <input type="checkbox"/> Remote Operator (ION, RO) and Army Cyber Operator (ACO) <input type="checkbox"/> Mission Commander (MC) <input type="checkbox"/> Exploitation Analyst (EA) <input type="checkbox"/> Digital Network Exploitation Analyst (DNEA) and Target Digital Network Analyst (TDNA) <input type="checkbox"/> Operational Target Development Analyst (OTDA), Targeteer, and Fire Support Planner <input type="checkbox"/> Cyber Operations Planner <input type="checkbox"/> Team Lead and Deputy Team Lead <input type="checkbox"/> Language Analyst (LA) <input type="checkbox"/> Target Analyst Reporter (TAR)	Trainee	Trainer	Date
Condition B – Standard A			

Trainer comments:

3.4.5 (CUI) Describe the USCC Intelligence Oversight Plan	Trainee	Trainer	Date
	Condition B – Standard A		

3.5 Mission Support Orientation

3.5.1 Describe and understand CSD fundamentals: <input type="checkbox"/> Mission and vision <input type="checkbox"/> Organizational hierarchy and task organization <input type="checkbox"/> CSD site model and stakeholder alignment <input type="checkbox"/> Work role training and certification model <input type="checkbox"/> Higher headquarters' mission and vision	Trainee	Trainer	Date
	Condition A – Standard A		
3.5.2 Describe CSD-supported organizations, including their purpose, location, and combatant command alignment: <input type="checkbox"/> JFHQ-C Army <input type="checkbox"/> JFHQ-C Air Force <input type="checkbox"/> JFHQ-C Navy <input type="checkbox"/> JFHQ-C Marines <input type="checkbox"/> JFHQ-DODIN <input type="checkbox"/> Cyber National Mission Force <input type="checkbox"/> Army Service <input type="checkbox"/> DCO support to Combatant Commands <input type="checkbox"/> CWB support to Army Service Component Commands <input type="checkbox"/> NETCOM, RCCs, and NECs	Trainee	Trainer	Date
	Condition B – Standard A		
3.5.3 Describe the purpose and resourcing of common CSD processes: <input type="checkbox"/> Operational requirements and essential components (e.g., stakeholder, validation, tool champion, timeline, etc.) <input type="checkbox"/> Requirements process concepts <input type="checkbox"/> Services required <input type="checkbox"/> Equipment required <input type="checkbox"/> Capability release requirements <input type="checkbox"/> Security practices	Trainee	Trainer	Date
	Condition A – Standard A		

Trainer comments:

3.6 Mission Process

3.6.1 (CUI) Describe the CSD standard of interaction between a capability development organization and its higher requirements-generating headquarters, including pre-requirement planning, drafting requirements, and interacting with tool champions, product owners, or requirement-owning stakeholders	Trainee	Trainer	Date
	Condition A – Standard A		
3.6.2 (CUI) Describe the U.S. Cyber Command testing and evaluation process, including Developmental Test and Evaluation, Developmental Acceptance, Evaluated Level of Assurance, and Operational Test and Evaluation	Trainee	Trainer	Date
	Condition A – Standard A		
3.6.3 Describe CSD capability development best practices <input type="checkbox"/> Naming convention methodology <input type="checkbox"/> Version control methodology <input type="checkbox"/> Integrated development environments and text editors used <input type="checkbox"/> Debugging tools used <input type="checkbox"/> Use of version control software and collaboration software	Trainee	Trainer	Date
	Condition A – Standard A		
3.6.4 Describe CSD documentation best practices <input type="checkbox"/> General internal <input type="checkbox"/> General external <input type="checkbox"/> Testing internal <input type="checkbox"/> Testing external <input type="checkbox"/> Developer internal	Trainee	Trainer	Date
	Condition A – Standard A		

Trainer comments:

4 (U) Module 4: Basic Cyberspace Capability Developer (CCD) Knowledge

4.1 C Programming

Training Resources: • *The C Programming Language (Kernighan, Ritchie)* • *Unix man pages* • *C Primer Plus (Prata)*

4.1.1 Describe the purpose and use of C programming fundamentals: <input type="checkbox"/> The <code>main()</code> function <input type="checkbox"/> The <code>return</code> statement <input type="checkbox"/> Macro guards <input type="checkbox"/> Data types <input type="checkbox"/> Functions and procedures <input type="checkbox"/> Parameters <input type="checkbox"/> Scope <input type="checkbox"/> Return values (return type and reference) <input type="checkbox"/> Header files <input type="checkbox"/> Keywords (static and extern) <input type="checkbox"/> Pointers <input type="checkbox"/> An array <input type="checkbox"/> C preprocessor <input type="checkbox"/> Casting <input type="checkbox"/> Control flow <input type="checkbox"/> Endianness <input type="checkbox"/> Multi-byte vs. Unicode character sets <input type="checkbox"/> Multi-threading <input type="checkbox"/> Hashing	Trainee	Trainer	Date
	Condition A – Standard A		
4.1.2 Describe C programming concepts in regards to memory: <input type="checkbox"/> Memory map of a C program <input type="checkbox"/> Stack <input type="checkbox"/> Heap <input type="checkbox"/> Stack vs Heap	Trainee	Trainer	Date
	Condition A – Standard A		

Trainer comments:

4.1.3 Demonstrate the proper declaration, understanding, and use of C data types and underlying structures: <input type="checkbox"/> char <input type="checkbox"/> short <input type="checkbox"/> int <input type="checkbox"/> long <input type="checkbox"/> long long <input type="checkbox"/> float <input type="checkbox"/> double <input type="checkbox"/> long double	Trainee	Trainer	Date
Condition C – Standard B			
4.1.4 Demonstrate proper declaration, understanding, and use of fixed-width C data types defined in stdint.h: <input type="checkbox"/> int8_t <input type="checkbox"/> uint8_t <input type="checkbox"/> int16_t <input type="checkbox"/> uint16_t <input type="checkbox"/> int32_t <input type="checkbox"/> uint32_t <input type="checkbox"/> int64_t <input type="checkbox"/> uint64_t	Trainee	Trainer	Date
Condition C – Standard B			
4.1.5 Demonstrate the ability to create and implement a function that uses different arrays: <input type="checkbox"/> An array <input type="checkbox"/> A multi-dimensional array	Trainee	Trainer	Date
Condition C – Standard B			

Trainer comments:

4.1.6 Demonstrate the ability to perform basic arithmetic operations using appropriate C operators while ensuring proper order of operations (PEMDAS): <input type="checkbox"/> Addition <input type="checkbox"/> Subtraction <input type="checkbox"/> Multiplication <input type="checkbox"/> Division <input type="checkbox"/> Modulus (%) <input type="checkbox"/> Pre-Increment (++i) <input type="checkbox"/> Post-Increment (i++) <input type="checkbox"/> Pre-Decrement (--i) <input type="checkbox"/> Post-Decrement (i--)	Trainee	Trainer	Date
	Condition C – Standard B		
	Trainee	Trainer	Date
4.1.7 Demonstrate the ability to properly use the standard main() entry arguments: <input type="checkbox"/> int argc <input type="checkbox"/> char *argv[]	Condition C – Standard B		
4.1.8 Demonstrate the ability to perform file management operations in C: <input type="checkbox"/> Open an existing file <input type="checkbox"/> Read data from a file <input type="checkbox"/> Write data to a file <input type="checkbox"/> Modify data in a file <input type="checkbox"/> Close an open file <input type="checkbox"/> Print file information to the console <input type="checkbox"/> Create a new file <input type="checkbox"/> Append data to an existing file <input type="checkbox"/> Delete a file <input type="checkbox"/> Determine the size of a file (in a UNIX-based operating system) <input type="checkbox"/> Determine location within a file <input type="checkbox"/> Insert data into an existing file	Trainee	Trainer	Date
	Condition C – Standard B		

Trainer comments:

4.1.9 Demonstrate the ability to create and implement functions to meet a requirement: <input type="checkbox"/> Proper declaration for created functions <input type="checkbox"/> A function that does not return a value (i.e., is declared void) <input type="checkbox"/> A function that is passed an argument by value <input type="checkbox"/> A function that takes a pointer argument <input type="checkbox"/> A function that returns a value using a return statement <input type="checkbox"/> A function that modifies an output parameter through a pointer <input type="checkbox"/> A function that receives input from a user <input type="checkbox"/> A function pointer <input type="checkbox"/> A recursive function	Trainee	Trainer	Date
Condition C – Standard B			
4.1.10 Demonstrate the ability to perform data validation: <input type="checkbox"/> Validating input receives matches input expected	Trainee	Trainer	Date
Condition C – Standard B			
4.1.11 Demonstrate skill in using pointers: <input type="checkbox"/> Declaring an integer pointer <input type="checkbox"/> Dereferencing a variable to get its value <input type="checkbox"/> Printing the address of the variable <input type="checkbox"/> Assigning a value to a pointer <input type="checkbox"/> Make use of a function pointer to call another function <input type="checkbox"/> Make effective use of pointer arithmetic to traverse an array	Trainee	Trainer	Date
Condition C – Standard B			
4.1.12 Demonstrate skill in creating and implementing conditional statements, expressions, and constructs: <input type="checkbox"/> for loop <input type="checkbox"/> while loop <input type="checkbox"/> do while loop <input type="checkbox"/> if statement <input type="checkbox"/> if/else statement <input type="checkbox"/> if/else if/else statement <input type="checkbox"/> switch statement <input type="checkbox"/> effective use of goto labels to construct a single exit point within a function	Trainee	Trainer	Date
Condition C – Standard B			

Trainer comments:

4.1.13 Demonstrate skill in using networking commands accounting for endianness: <input type="checkbox"/> socket() <input type="checkbox"/> send() <input type="checkbox"/> recv() <input type="checkbox"/> sendto() <input type="checkbox"/> recvfrom() <input type="checkbox"/> bind() <input type="checkbox"/> connect() <input type="checkbox"/> accept() <input type="checkbox"/> getsockopt() <input type="checkbox"/> setsockopt() <input type="checkbox"/> getaddrinfo() <input type="checkbox"/> gethostname() <input type="checkbox"/> struct sockaddr <input type="checkbox"/> struct sockaddr_in <input type="checkbox"/> struct sockaddr_un	Trainee	Trainer	Date
	Condition C – Standard B		
4.1.14 Demonstrate skill in creating and implementing a hash function	Trainee	Trainer	Date
	Condition C – Standard B		
4.1.15 Demonstrate skill in creating and implementing a sort routine	Trainee	Trainer	Date
	Condition C – Standard B		
4.1.16 Demonstrate skill in creating and implementing a state machine	Trainee	Trainer	Date
	Condition C – Standard B		
4.1.17 Describe terms associated with compiling, linking, debugging, and executables: <input type="checkbox"/> Portable Executable (PE) <input type="checkbox"/> Executable and Linkable Format (ELF) <input type="checkbox"/> Difference between PE and ELF <input type="checkbox"/> Difference between a library (shared object / DLL) and a regular executable program <input type="checkbox"/> Calling convention/Application Binary Interface (ABI)	Trainee	Trainer	Date
	Condition A – Standard A		

Trainer comments:

4.1.18 Demonstrate skill in compiling, linking, and debugging: <input type="checkbox"/> Execute a program in a debugger to perform general debugging actions <input type="checkbox"/> Create a program using the compilation and linking process <input type="checkbox"/> Compile position-independent code using a cross-compiler	Trainee	Trainer	Date
	Condition C – Standard B		
4.1.19 Demonstrate the ability to build a binary from multiple C source files and headers by writing a Makefile using explicit rules	Trainee	Trainer	Date
	Condition C – Standard B		
4.1.20 Describe how and when bitwise operators are used: <input type="checkbox"/> and (&) <input type="checkbox"/> or () <input type="checkbox"/> xor (^) <input type="checkbox"/> bitwise complement (~) <input type="checkbox"/> shift left (<<) <input type="checkbox"/> shift right (>>) <input type="checkbox"/> Add, removing, and testing for single-bit flags <input type="checkbox"/> Extracting arbitrary bytes from multi-byte data types	Trainee	Trainer	Date
	Condition C – Standard B		
4.1.21 Demonstrate skill in using the C preprocessor	Trainee	Trainer	Date
	Condition C – Standard B		
4.1.22 Demonstrate skill in accessing environment variables	Trainee	Trainer	Date
	Condition C – Standard B		
4.1.23 Demonstrate skill in controlling memory: <input type="checkbox"/> With attention given to implementation defined behavior, compare and contrast standard memory allocation functions (e.g., malloc(), calloc(), realloc(), and free()) <input type="checkbox"/> Demonstrate appropriate error checking when managing memory allocations <input type="checkbox"/> Describe programming techniques that reduce the occurrence of memory leaks (e.g., behaviors that reinforce a clear ownership model) <input type="checkbox"/> Demonstrate effective use of Valgrind with --leak-check=full to identify memory leaks <input type="checkbox"/> Given code samples, identify and remove memory leaks	Trainee	Trainer	Date
	Condition C – Standard B		

Trainer comments:

4.1.24 Describe the concepts and terminology associated with multi-threaded programs: <input type="checkbox"/> thread <input type="checkbox"/> pthread <input type="checkbox"/> fork <input type="checkbox"/> join <input type="checkbox"/> create <input type="checkbox"/> exit <input type="checkbox"/> detach <input type="checkbox"/> self <input type="checkbox"/> mutex <input type="checkbox"/> semaphore <input type="checkbox"/> race condition <input type="checkbox"/> deadlock <input type="checkbox"/> thread safe <input type="checkbox"/> thread id	Trainee	Trainer	Date
Condition A – Standard A			

4.1.25 Demonstrate the ability to manage memory in multi-threaded programs that make effective use of multithreaded programming constructs: <input type="checkbox"/> threads <input type="checkbox"/> semaphores <input type="checkbox"/> mutexes	Trainee	Trainer	Date
Condition C – Standard B			

Trainer comments:

4.2 Python Programming

Training Resources: • *Python 3 online documentation*

4.2.1 Describe purpose and use of foundational Python mechanics: <input type="checkbox"/> The return statement <input type="checkbox"/> Data types <input type="checkbox"/> A function <input type="checkbox"/> Parameters <input type="checkbox"/> Scope <input type="checkbox"/> Return values (return type and reference) <input type="checkbox"/> Import files <input type="checkbox"/> Dictionaries <input type="checkbox"/> List <input type="checkbox"/> Tuple <input type="checkbox"/> Singleton <input type="checkbox"/> The term mutable <input type="checkbox"/> The term immutable	Trainee	Trainer	Date
	Condition A – Standard A		
4.2.2 Demonstrate the proper declaration and use of Python data types and object-oriented constructs: <input type="checkbox"/> Integer (int) <input type="checkbox"/> Float (float) <input type="checkbox"/> String (str) <input type="checkbox"/> List (list) <input type="checkbox"/> Multi-dimensional list <input type="checkbox"/> Dictionary (dict) <input type="checkbox"/> Tuple (tuple) <input type="checkbox"/> Singleton	Trainee	Trainer	Date
	Condition C – Standard C		
4.2.3 Demonstrate the ability to perform basic arithmetic operations using Python operators while ensuring proper order of operations (PEMDAS): <input type="checkbox"/> Addition <input type="checkbox"/> Subtraction <input type="checkbox"/> Multiplication <input type="checkbox"/> Division <input type="checkbox"/> Modulus	Trainee	Trainer	Date
	Condition C – Standard C		

Trainer comments:

4.2.4 Demonstrate the ability to perform file management operations in Python: <input type="checkbox"/> Open an existing file <input type="checkbox"/> Read data from a file <input type="checkbox"/> Parse data from a file <input type="checkbox"/> Write data to a file <input type="checkbox"/> Modify data in a file <input type="checkbox"/> Close an open file <input type="checkbox"/> Print file information to the console <input type="checkbox"/> Create a new file <input type="checkbox"/> Append data to an existing file <input type="checkbox"/> Delete a file <input type="checkbox"/> Determine the size of a file <input type="checkbox"/> Determine location within a file <input type="checkbox"/> Insert data into an existing file	Trainee	Trainer	Date
	Condition C – Standard C		
4.2.5 Demonstrate the ability to create and implement functions to meet a requirement: <input type="checkbox"/> A function that returns multiple values <input type="checkbox"/> A function that receives input from a user <input type="checkbox"/> A recursive function	Trainee	Trainer	Date
	Condition C – Standard C		
4.2.6 Demonstrate the ability to perform data validation: <input type="checkbox"/> Validating receive input matches expected input <input type="checkbox"/> Creating a method for exception handling <input type="checkbox"/> Implementing a method for exception handling	Trainee	Trainer	Date
	Condition C – Standard C		
4.2.7 Demonstrate skill in creating and implementing conditional statements, expressions, and constructs: <input type="checkbox"/> for loop <input type="checkbox"/> while loop <input type="checkbox"/> with statement <input type="checkbox"/> if statement <input type="checkbox"/> if/else statement <input type="checkbox"/> if/elif/else statement	Trainee	Trainer	Date
	Condition C – Standard C		

Trainer comments:

4.2.8 Demonstrate skill in using networking commands accounting for endianness: <input type="checkbox"/> socket() <input type="checkbox"/> send() <input type="checkbox"/> recv() <input type="checkbox"/> sendto() <input type="checkbox"/> recvfrom() <input type="checkbox"/> bind() <input type="checkbox"/> listen() <input type="checkbox"/> connect() <input type="checkbox"/> accept() <input type="checkbox"/> close() <input type="checkbox"/> gethostname()	Trainee	Trainer	Date
	Condition C – Standard C		
4.2.9 Describe the terms and fundamentals associated with object-oriented programming using Python: <i>Training Resources: • Design Patterns: Elements of Reusable Object-Oriented Software (Gamma, Helm, Johnson, Vlissides)</i> <input type="checkbox"/> Class <input type="checkbox"/> Object <input type="checkbox"/> Difference between an object when discussing a class <input type="checkbox"/> Advantages to object-oriented programming <input type="checkbox"/> Inheritance <input type="checkbox"/> The keyword "super" <input type="checkbox"/> Initialization function of a constructor <input type="checkbox"/> The keyword "self" <input type="checkbox"/> The getter and setter functions <input type="checkbox"/> Attributes of a class <input type="checkbox"/> Factory design pattern <input type="checkbox"/> Singleton design pattern <input type="checkbox"/> Adapter design pattern <input type="checkbox"/> Bridge design pattern	Trainee	Trainer	Date
	Condition A – Standard A		
4.2.10 Demonstrate the ability to parse command line arguments using built-in functionality	Trainee	Trainer	Date
	Condition C – Standard C		

4.3 Data Structures

Training Resources: • Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles 5th edition (Narasimah Karumanchi)

Trainer comments:

4.3.1 Describe the concepts and terms associated with key data structures: <input type="checkbox"/> Binary search tree <input type="checkbox"/> Linked list <input type="checkbox"/> Double linked list <input type="checkbox"/> Circularly linked list <input type="checkbox"/> Weighted graph <input type="checkbox"/> Common pitfalls when using linked lists, trees, and graphs <input type="checkbox"/> The effect of First In First Out (FIFO) and Last In First Out (LIFO)	Trainee	Trainer	Date
Condition A – Standard A			
4.3.2 Demonstrate skill in creating and using a circularly linked list that accepts any data type: <input type="checkbox"/> Creating a circularly linked list with n number of items <input type="checkbox"/> Navigating through a circularly linked list <input type="checkbox"/> Finding the first occurrence of an item in a circularly linked list <input type="checkbox"/> Sorting the circularly linked list alphanumerically using a function pointer <input type="checkbox"/> Removing selected items from the circularly linked list <input type="checkbox"/> Inserting an item into a specific location in a circularly linked list <input type="checkbox"/> Removing all items from the circularly linked list <input type="checkbox"/> Destroying a circularly linked list	Trainee	Trainer	Date
Condition C – Standard B			
4.3.3 Demonstrate skill in creating and using a queue that accepts any data type: <input type="checkbox"/> Creating a queue with n number of items <input type="checkbox"/> Navigating through a queue to find the nth item <input type="checkbox"/> Finding an item in a queue <input type="checkbox"/> Removing selected items from a queue <input type="checkbox"/> Removing all items from the queue <input type="checkbox"/> Destroying a queue	Trainee	Trainer	Date
Condition C – Standard B			

Trainer comments:

4.3.4 Demonstrate skill in creating and using a tree that accepts any data type: <input type="checkbox"/> Creating a tree with n number of items <input type="checkbox"/> Navigating through a tree <input type="checkbox"/> Find the first occurrence of an item in a tree <input type="checkbox"/> Removing selected items from the tree <input type="checkbox"/> Inserting an item into a specified location in a tree <input type="checkbox"/> Removing all items from the tree <input type="checkbox"/> Destroying a tree	Trainee	Trainer	Date
	Condition C – Standard B		
4.3.5 Demonstrate skill in creating and using a binary search tree that accepts any data type: <input type="checkbox"/> Creating a binary search tree with n number of items <input type="checkbox"/> Navigating through a binary search tree <input type="checkbox"/> Locating an item in a binary search tree <input type="checkbox"/> Removing selected items from the binary search tree <input type="checkbox"/> Removing all items from the binary search tree <input type="checkbox"/> Describe implementation strategies for a balanced binary search tree <input type="checkbox"/> Destroying a binary search tree	Trainee	Trainer	Date
	Condition C – Standard B		
4.3.6 Demonstrate skill in creating and using a hash table that accepts any data type: <input type="checkbox"/> Creating a hash table with n number of items <input type="checkbox"/> Navigating through a hash table to find the nth item <input type="checkbox"/> Finding an item in a hash table <input type="checkbox"/> Removing selected items from a hash table <input type="checkbox"/> Inserting an item into a hash table <input type="checkbox"/> Implement functionality to mitigate hash collisions within the hash table <input type="checkbox"/> Removing all items from the hash table	Trainee	Trainer	Date
	Condition C – Standard B		

Trainer comments:

4.3.7 Demonstrate skill in creating and using a stack that accepts any data type: <input type="checkbox"/> Creating a stack with n number of items <input type="checkbox"/> Navigating through a stack to find the nth item <input type="checkbox"/> Adding an item in a stack <input type="checkbox"/> Removing selected items from a stack <input type="checkbox"/> Removing all items from the stack <input type="checkbox"/> Destroying a stack <input type="checkbox"/> Preventing a stack overrun	Trainee	Trainer	Date
	Condition C – Standard B		
4.3.8 Demonstrate skill in creating and using a weighted graph that accepts any data type: <input type="checkbox"/> Defining the structures required for graphs <input type="checkbox"/> Creating a graph with n number of nodes <input type="checkbox"/> Adding n number of edges to a graph <input type="checkbox"/> Finding a node within an existing graph <input type="checkbox"/> Finding an edge within a graph <input type="checkbox"/> Remove an edge from a graph <input type="checkbox"/> Remove a node and all of its edges from a graph <input type="checkbox"/> Calculate the weight of a path within a graph <input type="checkbox"/> Destroy the graph	Trainee	Trainer	Date
	Condition C – Standard B		
4.3.9 Demonstrate skill in implementing a priority queue that accepts any data type: <input type="checkbox"/> Defining the underlying structures required for priority queues <input type="checkbox"/> Assigning a priority to each element <input type="checkbox"/> Inserting an element into the priority queue <input type="checkbox"/> Removing the element with the highest priority from the priority queue <input type="checkbox"/> Destroying a priority queue <input type="checkbox"/> Define possible applications of a priority queue	Trainee	Trainer	Date
	Condition C – Standard B		

Trainer comments:

4.4 Algorithms

4.4.1 Demonstrate the ability to calculate runtime efficiency for a given algorithm using Asymptotic notation (Big-O) notation: <input type="checkbox"/> Insertion sort <input type="checkbox"/> Selection sort <input type="checkbox"/> Merge sort <input type="checkbox"/> Heap sort <input type="checkbox"/> Quick sort <input type="checkbox"/> State machine <input type="checkbox"/> Hashing	Trainee	Trainer	Date
	Condition B – Standard A		
4.4.2 Describe concepts associated with traversal techniques: <input type="checkbox"/> Depth first traversal <input type="checkbox"/> Breadth first traversal <input type="checkbox"/> The technique of determining the weight of a given path when traversing a graph <input type="checkbox"/> How the most efficient path for traversing a graph is determined	Trainee	Trainer	Date
	Condition B – Standard A		
4.4.3 Describe concepts associated with hashing: <input type="checkbox"/> Data distribution as it relates to hashing <input type="checkbox"/> Hash function efficiency <input type="checkbox"/> Hash collisions	Trainee	Trainer	Date
	Condition B – Standard A		
4.4.4 Demonstrate the ability to analyze sorting routines to determine the most efficient one to use, using an approximation of Big-O notation	Trainee	Trainer	Date
	Condition C – Standard A		

4.5 Operating System Concepts

Training Resources: • *Operating Systems: Three Easy Pieces (Arpaci-Dusseau)*

4.5.1 Describe terms and concepts associated with Operating System (OS) virtualization: <input type="checkbox"/> Processes <input type="checkbox"/> CPU scheduling <input type="checkbox"/> Paging tables <input type="checkbox"/> Caching <input type="checkbox"/> Kernel and user-mode memory	Trainee	Trainer	Date
	Condition B – Standard A		

Trainer comments:

4.5.2 Demonstrate the ability to use the following constructs associated with OS virtualization: <input type="checkbox"/> Interrupts <input type="checkbox"/> Signal handling	Trainee	Trainer	Date
Condition C – Standard B			
4.5.3 Describe terms and concepts associated with OS concurrency: <input type="checkbox"/> Threading <input type="checkbox"/> Locks <input type="checkbox"/> Race conditions <input type="checkbox"/> Deadlocks <input type="checkbox"/> Scheduling algorithms, i.e., round robin, shortest job first, priority scheduling, etc.	Trainee	Trainer	Date
Condition B – Standard A			
4.5.4 Demonstrate the ability to use the following constructs associated with OS concurrency: <input type="checkbox"/> Threads <input type="checkbox"/> Locks <input type="checkbox"/> Condition variables <input type="checkbox"/> Atomics	Trainee	Trainer	Date
Condition C – Standard B			
4.5.5 Describe terms and concepts associated with OS persistence: <input type="checkbox"/> Princeton / von Neumann architecture <input type="checkbox"/> Harvard architecture <input type="checkbox"/> File systems <input type="checkbox"/> The boot process	Trainee	Trainer	Date
Condition B – Standard A			

Trainer comments:

4.6 Secure Coding

Training Resources: • *Secure Coding in C and C++ 2nd edition (Robert Seacord)*

4.6.1 Describe terms and concepts associated with secure coding practices: <input type="checkbox"/> Common string-handling functions <input type="checkbox"/> Which functions guarantee null terminated strings <input type="checkbox"/> An off-by-one error <input type="checkbox"/> An integer overflow <input type="checkbox"/> A buffer overflow <input type="checkbox"/> The concept of use-after-free <input type="checkbox"/> Resource acquisition is initialization (RAII) <input type="checkbox"/> The difference between a regular expression and context-free grammar <input type="checkbox"/> The difference between input validation vs. input sanitization <input type="checkbox"/> The meaning of a pure function and if a function has a side-effect <input type="checkbox"/> General low-level crypto basics	Trainee	Trainer	Date
	Condition B – Standard A		
4.6.2 Demonstrate skill in using secure coding techniques: <input type="checkbox"/> Formatting string vulnerabilities <input type="checkbox"/> Safe buffer size allocation <input type="checkbox"/> Input sanitization <input type="checkbox"/> Input validation <input type="checkbox"/> Modeling complex functionality as state-machines <input type="checkbox"/> Establish a secure communications channel using an SSL library <input type="checkbox"/> Securely zeroing-out memory (compiler optimizations)	Trainee	Trainer	Date
	Condition C – Standard B		

4.7 Networking Fundamentals

Training Resources: • *Beej's guide to internet programming using internet sockets (Jorgensen)* • *Computer Networking: A Top-Down Approach 7th edition (James Kurose)*

4.7.1 Describe the concepts and terms associated with networking fundamentals: <input type="checkbox"/> Transmission Control Protocol (TCP) / User Datagram Protocol (UDP) <input type="checkbox"/> Open Systems Interconnect (OSI) model <input type="checkbox"/> POSIX API/BSD sockets <input type="checkbox"/> Purpose and use of sockets <input type="checkbox"/> Request For Comments (RFCs) <input type="checkbox"/> Purpose of subnetting	Trainee	Trainer	Date
	Condition B – Standard A		

Trainer comments:

4.7.2 Describe the concepts and terms associated with common protocols and their associated ports, if applicable: <input type="checkbox"/> Address Resolution Protocol (ARP) <input type="checkbox"/> Hypertext Transfer Protocol/Secure (HTTP/HTTPS) <input type="checkbox"/> Domain Name System (DNS) <input type="checkbox"/> Simple Mail Transfer Protocol (SMTP) <input type="checkbox"/> Internet Control Message Protocol (ICMP) <input type="checkbox"/> Dynamic Host Configuration Protocol (DHCP) <input type="checkbox"/> Internet Protocol version 4 (IPv4) <input type="checkbox"/> Internet Protocol version 6 (IPv6)	Trainee	Trainer	Date
	Condition B – Standard A		
4.7.3 Describe the addressing associated with key networking protocols: <input type="checkbox"/> IPv4 <input type="checkbox"/> IPv6 <input type="checkbox"/> Ethernet	Trainee	Trainer	Date
	Condition B – Standard A		

4.8 Serialization

4.8.1 Demonstrate the ability to handle partial reads and writes during serialization and de-serialization	Trainee	Trainer	Date
	Condition C – Standard B		
4.8.2 Demonstrate the ability to serialize fixed size multi-byte types between systems of differing endianness	Trainee	Trainer	Date
	Condition C – Standard B		
4.8.3 Demonstrate the ability to serialize and de-serialize variable sized data structures between systems of differing endianness	Trainee	Trainer	Date
	Condition C – Standard B		
4.8.4 Describe libraries commonly used to aid in serialization	Trainee	Trainer	Date
	Condition B – Standard A		

Trainer comments:

5 (U) Module 5: Army-specific operational training requirements and recommendations

This module covers any Army requirements and recommendations not covered in the preceding modules

5.1 There are no additional items in this module

Trainer comments:

6 (U) Module 6: Record of Assessments and Formal Training Completion

6.1 Basic Skill Level Exam

6.1.1 With references and required resources, pass the provided Basic Skill Level Exam to demonstrate proficiency as a basic skill level	Trainee	Trainer	Date
	Condition A – Standard A		

6.2 Basic Skill Level Validation Panel

6.2.1 With references and required resources having passed the basic skill level exam, pass the associated validation panel to demonstrate a firm understanding of the work submitted	Trainee	Trainer	Date
	Condition A – Standard A		

Trainer comments:

7 (U) Appendix A: Glossary

Term	Definition
CSD	Cyber Solutions Development
CWB	Cyber Warfare Battalion
CCD	Cyberspace Capability Developer
CNODP	Computer Network Operations Development Program
DDS	Defense Digital Service
ECT	Expeditionary Cyberspace Team
EW	Electronic Warfare
JQR	Job Qualification Record
MTE	Mentee/trainee
RF	Radiofrequency
TDQC	Tool Developer Qualification Course
TRNE	Mentee/trainee
TRR	Mentor/trainer

Trainer comments: