

UNCLASSIFIED//FOR OFFICIAL USE ONLY

JOB QUALIFICATION RECORD (JQR)
FOR
CYBERSPACE CAPABILITY
DEVELOPER (CCD)
BASIC LEVEL

TRINEE (TRNE) NAME: _____ SIGNATURE: _____
TRAINER (TRR) NAME: _____ SIGNATURE: _____
ASSIGN DATE: _____
ESTIMATED COMPLETION DATE: _____
JQR FINAL COMPLETION DATE: _____

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

UNCLASSIFIED//FOR OFFICIAL USE ONLY

TABLE OF CONTENTS

(U) CONTENTS.....	4
(U) PROFICIENCY/WORK FUNCTION.....	4
(U) PREREQUISITES & REQUIREMENTS	4
(U) WORK DESCRIPTION	5
(U) SKILL PROGRESSION ROADMAP	5
(U) CONDITIONS AND STANDARDS	6
(U) MODULE 1.0 – ADMINISTRATIVE	8
(U) Accounts: Network and Database	8
(U) Security Read-Ins	8
(U) Network Permissions/Groups	8
(U) Formal Accesses	8
(U) Mission Element Required Training (Common Core Training)	8
(U) JQR Required Training (Mission Specific)	8
(U) JQR Recommended Training (Mission Specific).....	8
(U) Required Certifications	8
(U) Recommended Certifications	8
(U) Pipeline Training.....	9
(U) MODULE 2.0 – ORGANIZATIONAL PRIMER.....	10
(U//FOUO) Mission and Organizational Partners	10
(U) Policy and Compliance	10
(U) Mission Support Orientation	11
(U) Mission Process.....	12
(U) MODULE 3.0 – CYBERSPACE CAPABILITY DEVELOPER (CCD) KNOWLEDGE	13
3.1 (U) C Programming	13
3.2 (U) Python Programming	17
3.3 (U) Data Structures	20
3.4 (U) Algorithms	23
3.5 (U) Operating System Concepts	24
3.6 (U) Secure Coding	25
3.7 (U) Networking Fundamentals	26
3.8 (U) Serialization.....	26
3.9 (U) Basic Exam	27
3.10 (U) Basic Validation Panel	27
(U) APPENDIX A – JQR GLOSSARY.....	28
(U) APPENDIX B – REFERENCE MATERIALS	31
(U) APPENDIX C – CYBERSPACE CAPABILITIES DEVELOPER (CCD) KNOWLEDGE MATRIX.....	32
(U) APPENDIX D – MODULE SIGNATURE PAGE	33

UNCLASSIFIED//FOR OFFICIAL USE ONLY

THIS PAGE INTENTIONALLY LEFT BLANK

UNCLASSIFIED//FOR OFFICIAL USE ONLY

****Applicable to USCYBERCOM Title 10 Mission Authorities****

(U) CONTENTS

(U//FOUO) This JQR is divided into 3 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 team must know. The 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 final JQR qualification, and the information on the tools, databases, and systems required to conduct the mission.

(U) PROFICIENCY/WORK FUNCTION

(U//FOUO) The purpose of this Job Qualification Record (JQR) is to train the individual on the knowledge, skills and abilities necessary to perform at **Basic Proficiency as a Cyberspace Capability Developer (CCD)** for all assigned (civilian, military and contract) personnel within the development organization assigned to **Cyber Solutions Detachment (CSD)**, **Cyber Warfare Support Battalion (CWSB)**, **Cyber Protection Brigade (CPB)** or any other Army organization recognized as providing cyberspace capability development.

(U) PREREQUISITES & REQUIREMENTS

(U) JQR Prerequisites:

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

- Degree from an accredited institution in Computer Science, Electrical Engineering, or related field
- Graduation from the Tool Developer Qualification Course (TDQC) or its derivatives
- Documented work experience

(U) All prospects that did not graduate from the 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.

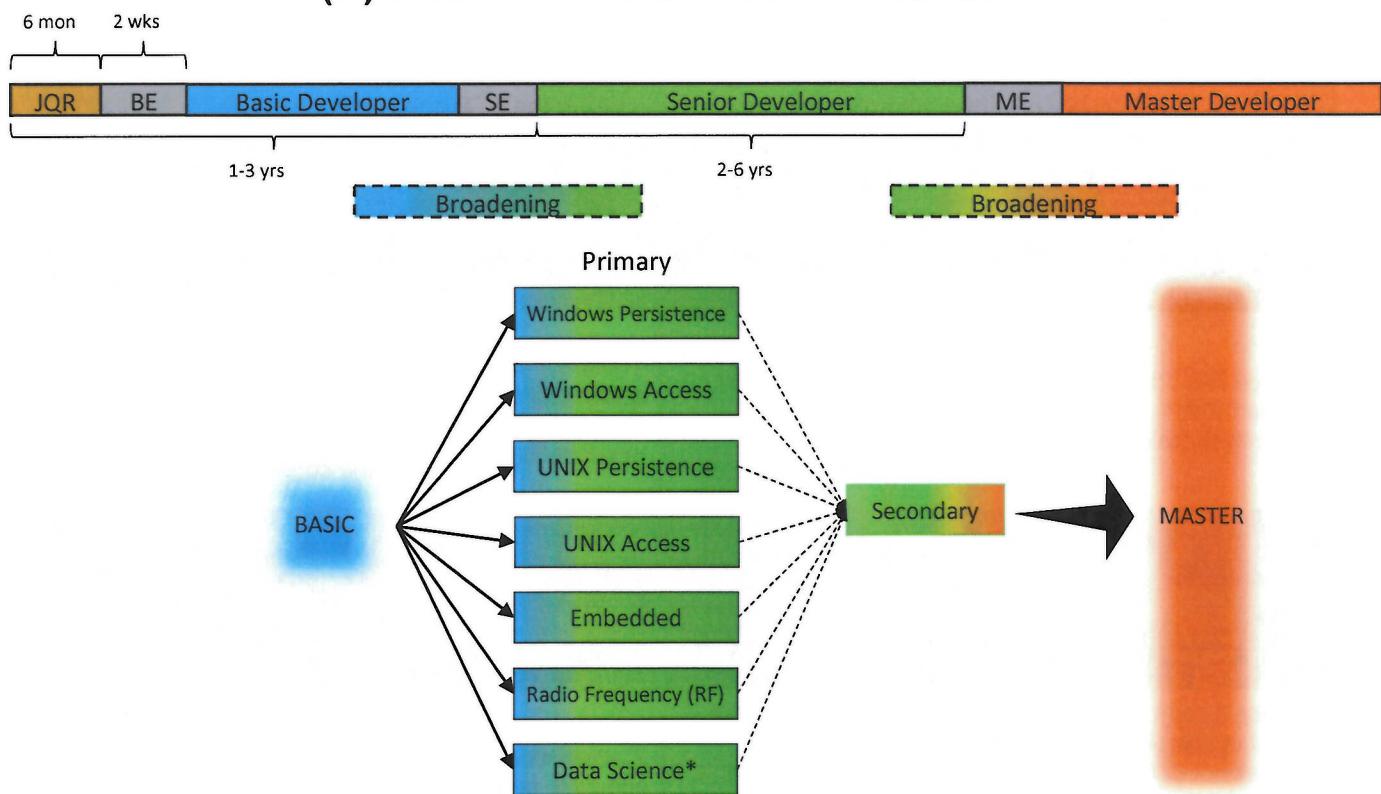
(U) JQR Requirements:

(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 Basic exam.
- Receive Basic Certification recommendation from validation panel.

(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. The CCD's primary objective is to provide capabilities that produce an effect in and throughout cyberspace operations. 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.

(U) SKILL PROGRESSION ROADMAP

(U) CCD's advance in skill and ability through mentorship and experience as they progress and specialize in three skill levels (Basic, Senior, and Master). 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 CNODP, TWI, USMA Instructor, Researcher, DDS AEP, ACS, and branch-specific PME. Additionally, prior to master skill level, a CCD must certify as a senior in secondary specialty.

(U) CONDITIONS AND STANDARDS

(U) Below are the conditions and standards that are applied to the JQR.

CONDITIONS – These are the conditions that are referenced by the CON column in the JQR	
CODE	CONDITIONS
A	(U//FOUO) The individual will complete the JQR tasks without reference and resources.
B	(U//FOUO) The individual will complete the JQR tasks with open-source resources, and personal references and resources in both a written and digital medium.
C	(U//FOUO) The individual will complete the JQR tasks with open-source and personal references and resources, both written and digital, as well as a workstation with access to necessary tools and software

STANDARDS – These are the standards that are referenced by the STD column in the JQR				
CODE: A	In accordance with the below scale below demonstrate knowledge related to the associated task			
	Needs Retraining		Meets Standard	Exceeds Standard
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 term or task 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 shows advanced comprehension in the cognitive domain.
CODE B:	In accordance with the scale below demonstrate the ability to complete the specified task with compiled C code			
	Needs Retraining		Meets Standard	Exceeds Standard
0	1	2	3	4
Trainee's code 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 it 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.
CODE: C	In accordance with the scale below demonstrate the ability to complete the specified task with python code			
	Needs Retraining		Meets Standard	Exceeds Standard
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 the code does not address the task.	Trainee writes code that executes without syntax errors and is related to the task but	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.

UNCLASSIFIED//FOR OFFICIAL USE ONLY

		contains runtime errors.		
CODE: D	Task completed to organizational standards, all services and accounts related to the task are functional and operate as expected			
Needs Retraining		0	Meets Standard	Exceeds Standard
			3	4
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.	

(U) MODULE 1.0 – ADMINISTRATIVE**(U) Module 1.0 - covers all initial requirements that must be met prior to completion of the JQR.**

Item #	Knowledge, Skill, or Ability	CON	STD	TRR	SCORE	MTE	DATE
1.0	(U) Accounts: Network and Database						
1.0.1	(U//FOUO) Accounts: a. Top Secret/Sensitive b. Compartmented Information (SCI) c. Public Key Infrastructure (PKI) d. Rapid Capability Developer Network (RCDN)	B	D				
1.1	(U) Security Read-Ins						
1.1.1	(U//FOUO) Security Read-Ins: a. SI b. Talent Keyhole (TK) c. As required by organization	B	D				
1.2	(U) Network Permissions/Groups						
1.2.1	(U//FOUO) Network Permissions/Groups: a. As required by organization	B	D				
1.3	(U) Formal Accesses						
1.3.1	(U) Formal Accesses: a. As required by organization	B	D				
1.4	(U) Mission Element Required Training (Common Core Training)						
1.4.1	(U) Mission Element Required Training: a. None	B	D				
1.5	(U) JQR Required Training (Mission Specific)						
1.5.1	(U) JQR Required Training: a. None	NA	NA				
1.6	(U) JQR Recommended Training (Mission Specific)						
1.6.1	(U) JQR Recommended Training: a. Read and understand JP 3-12.	B	D				
1.7	(U) Required Certifications						
1.7.1	(U) Required Certification: a. None	NA	NA				
1.8	(U) Recommended Certifications						
1.8.1	(U) Recommended Certification: a. None	NA	NA				

UNCLASSIFIED//FOR OFFICIAL USE ONLY

1.9	(U) Pipeline Training							
1.9.1	(U) Pipeline Training: a. None	NA	NA					

(U) MODULE 2.0 – ORGANIZATIONAL PRIMER

Item #	Knowledge, Skill, or Ability	CON	STD	TRR	SCORE	MTE	DATE
2.0	(U//FOUO) Mission and Organizational Partners						
2.0.1	(U) Describe the mission of the Cyber Mission Force (CMF).	A	A				
2.0.2	(U) Describe the mission of the CMF team types: a. Combat Mission Team (CMT) b. National Mission Team (NMT) c. Combat Support Team (CST) d. National Support Team (NST) e. Cyber Protection Team (CPT)	A	A				
2.0.3	(U) Describe the different work roles on the CMF teams. TR: USCC CFCOE	B	A				
2.0.4	(U) Describe the mission and responsibilities of the following organizations: a. USCC b. Defense Threat Reduction Agency (DTRA) c. Office of the Secretary of Defense, d. Director Operational Test and Evaluation (DOT&E) f. Federally Funded Research and Development Centers (FFRDC) h. Central Intelligence Agency (CIA) i. Open Source Center j. Federal Bureau of Investigation (FBI) k. Department of Homeland Security (DHS) l. Defense Digital Services (DDS) m. Army Research Lab (ARL) n. National Security Agency (NSA) o. National Air and Space Intelligence Center (NASIC) p. TRADOC Capability Manager (TCM-Cyber)	B	A				
2.1	(U) Policy and Compliance						
2.1.1	(U) Demonstrate knowledge of U.S. Codes and their application to the Intelligence Community (IC) and Cyberspace Operations (Title 10, 18, and 50)	B	A				
2.1.2	(U) Describe DoD Directive 5240.1R to include Questionable Intelligence Activities.	B	A				

UNCLASSIFIED//FOR OFFICIAL USE ONLY

2.1.3	(U) Describe DoD Directive 5240.1.	B	A				
2.1.4	(U) Describe NSA/CSS Policy 11-1.	B	A				
2.1.5	(U) Describe the FISA act of 1978.	B	A				
2.1.6	(U//FOUO) Describe USSID CR1610.	B	A				
2.1.7	(U//FOUO) Demonstrate the ability to properly use classification marking and handling caveats.	B	A				
2.1.8	(U//FOUO) Demonstrate the ability to properly store and forward information with handling caveats.	B	D				
2.1.9	(U//FOUO) Describe USSID SP0018 to include violations and reporting.	B	A				
2.1.10	(U//FOUO) Describe Title 10 and its importance.	B	A				
2.1.11	(U//FOUO) Describe Title 50 and its importance.	B	A				
2.1.12	(U//FOUO) Describe the difference between conducting cyberspace intelligence, surveillance, and reconnaissance (C-ISR) and cyberspace surveillance and reconnaissance (C-SR).	B	A				
2.1.13	(U//FOUO) Describe the USCC Intelligence Oversight Plan.	B	A				
2.1.14	(U//FOUO) Demonstrate working knowledge of the policies, laws, and the authorities to operate that govern capability development.	B	A				
2.2	(U) Mission Support Orientation						
2.2.1	(U) Describe the organizational hierarchy of the organization: a. Organization's mission and vision b. Organizational hierarchy c. Higher headquarters' mission and vision	A	A				
2.2.2	(U) Describe the organizations that are supported within the organization: a. Supported organizations b. Supporting organizations Requirements process	A	A				
2.2.3	(U) Describe the operations process within the organization: a. Operational requirements b. Services required	A	A				

UNCLASSIFIED//FOR OFFICIAL USE ONLY

	c. Equipment required d. Capability release requirements e. Security practices						
2.3	(U) Mission Process						
2.3.1	(U) Describe your organization's development best practices within the organizations framework: a. Naming convention methodology b. Version control methodology c. Integrated development environments and text editors used d. Debugging tools used e. Use of version control software (Gitlab, Git, Bitbucket, or the Atlassian stack) and collaboration software (Gitlab, confluence or Jira for example) within the organization.	A	A				
2.3.2	(U) Describe your organization's documentation process: a. General internal b. General external c. Testing internal d. Testing external e. Developer internal	A	A				

(U) MODULE 3.0 – CYBERSPACE CAPABILITY DEVELOPER (CCD) KNOWLEDGE

Item #	Knowledge, Skill, or Ability	CON	STD	TRR	SCORE	MTE	DAT E
3.1 (U) C Programming							
(U) Knowledge references for this section can be found in Appendix B-1							
3.1.1	(U) Describe purpose and use of: <ul style="list-style-type: none"> a. The main() function b. The return statement c. Macro guards d. Data types e. Functions and procedures f. Parameters g. Scope h. Return values (return type and reference) i. Header files j. Keywords (static and extern) k. Pointers l. An array m. C preprocessor n. Casting o. Control flow p. Endianness q. Multi-byte vs. Unicode character sets r. Multi-threading s. Hashing 	A	A				
3.1.2	(U) Describe the C programming concepts in regards to: <ul style="list-style-type: none"> a. Memory map of a C program b. Stack c. Heap d. Stack vs Heap 	A	A				
3.1.3	(U) Demonstrate the proper declaration, understanding, and use of C data types and underlying structures: <ul style="list-style-type: none"> a. WORD b. DWORD c. QUADWORD d. Short e. Integer (int) f. Float (float) g. Character (char) h. Double (double) i. Long (long) 	C	B				

UNCLASSIFIED//FOR OFFICIAL USE ONLY

3.1.4	(U) Demonstrate the ability to create and implement a function that uses different arrays: a. An array b. A multi-dimensional array	C	B				
3.1.5	(U) Demonstrate the ability to perform basic arithmetic operations using appropriate C operators while ensuring proper order of operations (PEMDAS): a. Addition (+) b. Subtraction (-) c. Multiplication (*) d. Division (/) e. Modulus (%) f. Increment (++) g. Decrement (--)	C	B				
3.1.6	(U) Demonstrate the ability to properly use the standard main() entry arguments: a. int argc b. char **argv c. char **envp	C	B				
3.1.7	(U) Demonstrate the ability to perform file management operations in C: a. Open an existing file b. Read data from a file c. Write data to a file d. Modify data in a file e. Close an open file f. Print file information to the console g. Create a new file h. Append data to an existing file i. Delete a file j. Determine the size of a file k. Determine location within a file l. Insert data into an existing file	C	B				
3.1.8	(U) Demonstrate the ability to create and implement functions to meet a requirement: a. Proper declarations for created functions b. A function that returns a void c. A function that is passed an argument by value d. A function that is passed an argument by reference e. A function that returns a value using a return statement f. A function that returns a value by reference g. A function that receives input from a user h. A function pointer i. A recursive function	C	B				

UNCLASSIFIED//FOR OFFICIAL USE ONLY

3.1.9	(U) Demonstrate the ability to perform data validation: a. Validating input receives matches input expected	C	B				
3.1.10	(U) Demonstrate skill in using pointers: a. Declaring an integer pointer b. Dereferencing a variable to get its value c. Printing the address of the variable d. Assigning a value to a pointer e. Make use of a function pointer to call another function f. Make effective use of pointer arithmetic to traverse an array	C	B				
3.1.11	(U) Demonstrate skill in creating and implementing conditional statements, expressions, and constructs: a. for loop b. while loop c. do while loop d. if statement e. if->else statement f. if->else if->else statement g. switch statement h. effective use of goto labels to construct a single exit point within a function	C	B				
3.1.12	(U) Demonstrate skill in using networking commands accounting for endianness: a. socket() b. send() c. recv() d. sendto() e. recvfrom() f. bind() g. connect() h. accept() i. getsockopt() j. setsockopt() k. getaddrinfo() l. gethostname m. sethostname n. struct sockaddr{} o. struct sockaddr_in{} p. struct sockaddr_un{}	C	B				

UNCLASSIFIED//FOR OFFICIAL USE ONLY

	TR: See Appendix B-3					
3.1.13	(U) Demonstrate skill in creating and implementing a hash function.	C	B			
3.1.14	(U) Demonstrate skill in creating and implementing a sort routine.	C	B			
3.1.15	(U) Demonstrate skill in creating and implementing a state machine.	C	B			
3.1.16	(U) Demonstrate skill in implementing Dijkstra's Algorithm to find the optimal path in a weighted graph.	C	B			
3.1.17	(U) Describe terms associated with compiling, linking, debugging, and executables: a. Difference between PE and ELF b. Portable Executable (PE) c. Executable and Linkable Format (ELF) d. Difference between a library (shared object / DLL) and a regular executable program e. Calling convention/Application Binary Interface (ABI)	C	B			
3.1.18	(U) Demonstrate skill in compiling, linking, and debugging a. Execute a program in a debugger to perform general debugging actions b. Create a program using the compilation and linking process c. Compile position-independent code using a cross-compiler	B	A			
3.1.19	(U) Describe how and when bitwise operators are used a. and (&) b. or () c. xor (^) d. bitwise complement (~) e. shift left (<<) f. shift right (>>)	C	B			
3.1.20	(U) Demonstrate skill in using bitwise operator a. and (&) b. or () c. xor (^) d. bitwise complement (~) e. shift left (<<) f. shift right (>>)	C	B			
3.1.21	(U) Demonstrate skill in using the C preprocessor	C	B			
3.1.22	(U) Demonstrate skill in controlling memory: a. Identify memory leaks b. Remove identified memory leaks	B	B			

UNCLASSIFIED//FOR OFFICIAL USE ONLY

	c. Make effective use of Valgrind with --leak-check=full to identify and correct memory leaks and context errors						
	(U) Describe the concepts and terminology associated with multi-threaded programs:						
3.1.23	a. thread	C					
	b. pthread	C					
	c. fork	C					
	d. join	C					
	e. create	C					
	f. exit	C					
	g. detach	C					
	h. self	C					
	i. mutex	C					
	j. semaphore	C					
	k. race condition	C					
	l. deadlock	C					
	m. thread safe	C					
	n. thread id	C					
3.1.24	(U) Demonstrate the ability to manage memory in multi-threaded programs that make effective use of multithreaded programming constructs to include but not limited to:						
	a. threads	C	B				
	b. semaphores	C	B				
	c. mutexes	C	B				
3.2 (U) Python Programming							
(U) Knowledge references for this section can be found in Appendix B-2							
3.2.1	(U) Describe purpose and use of:						
	a. The return statement	B	C				
	b. Data types	B	C				
	c. A function	B	C				
	d. Parameters	B	C				
	e. Scope	B	C				
	f. Return values (return type and reference)	B	C				
	g. Import files	B	C				
	h. Dictionaries	B	C				

UNCLASSIFIED//FOR OFFICIAL USE ONLY

	i. List	B	C				
	j. Tuple	B	C				
	k. Structs	B	C				
	l. Unions	B	C				
	m. Singleton	B	C				
	n. The term mutable	B	C				
	o. The term immutable	B	C				
3.2.2	(U) Demonstrate the proper declaration and use of Python data types and object-oriented constructs:						
	a. Integer (int)	C	C				
	b. Float (float)	C	C				
	c. String (str)	C	C				
	d. List (list)	C	C				
	e. Multi-dimensional list	C	C				
	f. Dictionary (dict)	C	C				
	g. Tuple (tuple)	C	C				
	h. Singleton	C	C				
3.2.3	(U) Demonstrate the ability to perform basic arithmetic operations using appropriate Python operators while ensuring proper order of operations (PEMDAS):						
	a. Addition (+)	C	C				
	b. Subtraction (-)	C	C				
	c. Multiplication (*)	C	C				
	d. Division (/)	C	C				
	e. Modulus (%)	C	C				
3.2.4	(U) Demonstrate the ability to perform file management operations in Python:						
	a. Open an existing file	C	C				
	b. Read data from a file	C	C				
	c. Parse data from a file	C	C				
	d. Write data to a file	C	C				
	e. Modify data in a file	C	C				
	f. Close an open file	C	C				
	g. Print file information to the console	C	C				
	h. Create a new file	C	C				
	i. Append data to an existing file	C	C				

UNCLASSIFIED//FOR OFFICIAL USE ONLY

	j. Delete a file	C	C				
	k. Determine the size of a file	C	C				
	l. Determine location within a file	C	C				
	m. Insert data into an existing file	C	C				
3.2.5	(U) Demonstrate the ability to create and implement functions to meet a requirement:						
	a. A function that returns multiple values	C	C				
	b. A function that receives input from a user	C	C				
	c. A recursive function	C	C				
3.2.6	(U) Demonstrate the ability to perform data validation:						
	a. Validating received input matches expected input	C	C				
	b. Creating a method for exception handling	C	C				
	c. Implementing a method for exception handling	C	C				
3.2.7	(U) Demonstrate skill in creating and implementing conditional statements, expressions, and constructs:						
	a. for loop	C	C				
	b. while loop	C	C				
	c. if statement	C	C				
	d. if->else statement	C	C				
	e. if->elif->else statement	C	C				
3.2.8	(U) Demonstrate skill in using networking commands accounting for endianness: See Appendix B-3						
	a. socket()	C	C				
	b. send()	C	C				
	c. recv()	C	C				
	d. sendto()	C	C				
	e. recvfrom()	C	C				
	f. bind()	C	C				
	g. listen()	C	C				
	h. connect()	C	C				
	i. accept()	C	C				
	j. close()	C	C				
	k. gethostname()	C	C				

UNCLASSIFIED//FOR OFFICIAL USE ONLY

	(U) Describe the terms and fundamentals associated with object-oriented programming using Python: See Appendix B-5						
3.2.9	a. Class	B	A				
	b. Object	B	A				
	c. Difference between an object when discussing a class	B	A				
	d. Advantages to object-oriented programming	B	A				
	e. Inheritance	B	A				
	f. The keyword "Super"	B	A				
	g. Initialization function of a constructor	B	A				
	h. The keyword "self"	B	A				
	i. The getter and setter functions	B	A				
	j. Attributes of a class	B	A				
	k. Factory design pattern	B	A				
	l. Singleton design pattern	B	A				
	m. Adapter design pattern	B	A				
	n. Bridge design pattern	B	A				
3.2.10	(U) Demonstrate the ability to parse command line arguments using built-in functionality	B	C				

3.3 (U) Data Structures

(U) Knowledge references for this section can be found in Appendix B-4

	(U) Describe the concepts and terms associated with the following data structures:						
3.3.1	a. Binary search tree	B	A				
	b. Linked list	B	A				
	c. Doubly linked list	B	A				
	d. Circularly linked list	B	A				
	e. Weighted Graph	B	A				
	f. Common pitfalls when using linked lists, trees, and graphs	B	A				
	g. The effect of First in first out (FIFO) and Last in first out (LIFO)	B	A				
	(U) Demonstrate skill in creating and using a linked list:						
3.3.2	a. Creating a linked list with n number of items	B	C				
	b. Navigating through a linked list	B	C				

UNCLASSIFIED//FOR OFFICIAL USE ONLY

	c. Retrieving the first occurrence of an item in a linked list	B	C				
	d. Sorting the linked list alphanumerically using a function pointer	B	C				
	e. Removing selected items from the linked list	B	C				
	f. Inserting an item into a specified location in a linked list	B	C				
	g. Removing all items from the linked list	B	C				
	h. Destroying a linked list	B	C				
3.3.3	(U) Demonstrate skill in creating and using a doubly linked list:						
	a. Creating a doubly linked list with n number of items	B	C				
	b. Navigating through a doubly linked list	B	C				
	c. Finding the first occurrence of an item in a doubly linked list	B	C				
	d. Sorting the doubly linked list alphanumerically using a function pointer	B	C				
	e. Removing selected items from the doubly linked list	B	C				
	f. Inserting an item into a specified location in a doubly linked list	B	C				
	g. Removing all items from the doubly linked list	B	C				
	h. Destroying a doubly linked list	B	C				
	(U) Demonstrate skill in creating and using a circularly linked list:						
3.3.4	a. Creating a circularly linked list with n number of items	B	C				
	b. Navigating through a circularly linked list	B	C				
	c. Finding the first occurrence of an item in a circularly linked list	B	C				
	d. Sorting the circularly linked list alphanumerically using a function pointer	B	C				
	e. Removing selected items from the circularly linked list	B	C				

UNCLASSIFIED//FOR OFFICIAL USE ONLY

	f. Inserting an item into a specified location in a circularly linked list	B	C				
	g. Removing all items from the circularly linked list	B	C				
	h. Destroying a circularly linked list	B	C				
(U) Demonstrate skill in creating and using a queue:							
3.3.5	a. Creating a queue with n number of items	B	C				
	b. Navigating through a queue to find the nth item	B	C				
	c. Finding an item in a queue	B	C				
	d. Removing selected items from a queue	B	C				
	e. Removing all items from the queue	B	C				
	f. Destroying a queue	B	C				
(U) Demonstrate skill in creating and using a tree:							
3.3.6	a. Creating a tree with n number of items	B	C				
	b. Navigating through a tree	B	C				
	c. Finding the first occurrence of an item in a tree	B	C				
	d. Removing selected items from the tree	B	C				
	e. Inserting an item into a specified location in a tree	B	C				
	f. Removing all items from the tree	B	C				
	g. Destroying a tree	B	C				
(U) Demonstrate skill in creating and using a binary search tree:							
3.3.7	a. Creating a binary search tree with n number of items	B	C				
	b. Navigating through a binary search tree	B	C				
	c. Locating an item in a binary tree	B	C				
	d. Removing selected items from the binary search tree	B	C				
	e. Removing all items from the binary search tree	B	C				
	f. Describe implementation strategies for a balanced binary search tree	B	C				
	g. Destroying a binary search tree	B	C				

UNCLASSIFIED//FOR OFFICIAL USE ONLY

3.3.8	(U) Demonstrate skill in creating and using a hash table:						
	a. Creating a hash table with n number of items	B	C				
	b. Navigating through a hash table to find the nth item	B	C				
	c. Finding an item in a hash table	B	C				
	d. Removing selected items from a hash table	B	C				
	e. Inserting an item into a hash table	B	C				
	f. Implement functionality to mitigate hash collisions within the hash table	B	C				
	g. Removing all items from the hash table	B	C				
3.3.9	(U) Demonstrate skill in creating and using a stack:						
	a. Creating a stack with n number of items	B	C				
	b. Navigating through a stack to find the nth item	B	C				
	c. Adding an item in a stack	B	C				
	d. Removing selected items from a stack	B	C				
	e. Removing all items from the stack	B	C				
	f. Destroying a stack	B	C				
	g. Preventing a stack overrun	B	C				
3.3.10	(U) Demonstrate skill in creating and using a weighted graph:						
	a. Defining the structures required for graphs	B	C				
	b. Creating a graph with n number of nodes	B	C				
	c. Adding n number of edges to a graph	B	C				
	d. Finding a node within an existing graph	B	C				
	e. Finding an edge within a graph	B	C				
	f. Remove an edge from a graph	B	C				
	g. Remove a node and all of its edges from a graph	B	C				
	h. Calculate the weight of a path within a graph	B	C				
	i. Destroy the graph	B	C				
3.4 (U) Algorithms							
(U) Knowledge references for this section can be found in Appendix B-4							
3.4.1	(U) Asymptotic notation (Big-O) Demonstrate the ability to calculate runtime efficiency for a given algorithm using Big-O notation.						

UNCLASSIFIED//FOR OFFICIAL USE ONLY

	Insertion sort	B	A				
	Selection sort	B	A				
	Merge sort	B	A				
	Heap sort	B	A				
	Quick sort	B	A				
	State machine	B	A				
	Hashing	B	A				
3.4.2	(U) Describe concepts associated with traversal techniques:						
	Depth first traversal	B	A				
	Breadth first traversal	B	A				
	The technique of determining the weight of a given path when traversing a graph	B	A				
	How the most efficient path for traversing a graph is determined	B	A				
3.4.3	(U) Describe concepts associated with hashing: Data distribution as it relates to hashing Hash function efficiency Hash collisions	B	A				
3.4.4	(U) Demonstrate the ability to analyze sorting routines to determine the most efficient one to use, using an approximation of Big-O notation.	C	A				

3.5 (U) Operating System Concepts

(U) Knowledge references for this section can be found in Appendix B-6

3.5.1	(U) Describe terms and concepts associated with OS Virtualization: a. Processes b. CPU scheduling c. Paging tables d. Caching e. Kernel and user-mode memory	B	A				
3.5.2	(U) Demonstrate the ability to use the following constructs associated with OS Virtualization: a. Interrupts b. Signal handling	C	B				
3.5.3	(U) Describe terms and concepts associated with OS Concurrency: a. Threading b. Locks c. Race conditions d. Deadlocks e. Scheduling modules i.e. round-robin, etc.	B	A				

UNCLASSIFIED//FOR OFFICIAL USE ONLY

3.5.4	(U) Demonstrate the ability to use the following constructs associated with OS Concurrency: a. Threads b. Locks	C	B					
3.5.5	(U) Describe terms and concepts associated with OS Persistence:							
	a. Von Nuemann architecture	B	A					
	b. Harvard architecture	B	A					
	c. File systems	B	A					
	d. The boot process	B	A					

3.6 (U) Secure Coding

(U) Knowledge references for this section can be found in Appendix B-7

3.6.1	(U) Describe terms and concepts associated with secure coding practices:							
	a. Common string-handling functions	B	A					
	b. Which functions guarantee null terminated strings	B	A					
	c. An off-by-one error	B	A					
	d. An integer overflow	B	A					
	e. A buffer overflow	B	A					
	f. The concept of use-after-free	B	A					
	g. Resource acquisition is initialization(RAI)	B	A					
	h. The difference between a regular expression and a context-free grammar	B	A					
	i. The difference between input validation vs. input sanitization	B	A					
	j. The meaning of a pure function and if a function has a side-effect	B	A					
	k. General low-level crypto basics	B	A					
3.6.2	(U) Demonstrate skill in using secure coding techniques:							
	Formatting string vulnerabilities	C	B					
	Safe buffer size allocation	C	B					
	Input sanitization	C	B					
	Input validation	C	B					
	Modeling complex functionality as state-machines	C	B					
	Establish a secure communications channel using an SSL library	C	B					
	Securely zeroing-out memory	C	B					

UNCLASSIFIED//FOR OFFICIAL USE ONLY

	(compiler optimizations)						
3.7 (U) Networking Fundamentals							
(U) Knowledge references for this section can be found in Appendix B-3							
3.7.1	(U) Describe the concepts and terms associated with the following networking fundamentals:						
	Transmission Control	B	A				
	Protocol (TCP) / datagrams (UDP)	B	A				
	Open Systems Interconnect (TCP/OSI) model	B	A				
	POSIX API/BSD sockets	B	A				
	Purpose and use of sockets	B	A				
	Request for comments (RFC)	B	A				
3.7.2	Purpose of sub-netting	B	A				
	(U) Describe the concepts and terms associated with common protocols and their associated ports, if applicable:						
	Address Resolution Protocol (ARP)	B	A				
	Hypertext Transfer Protocol/Secure (HTTP/HTTPS)	B	A				
	Domain Name System (DNS)	B	A				
	Simple Mail Transfer Protocol (SMTP)	B	A				
	Internet Control Message Protocol (ICMP)	B	A				
	Dynamic Host Configuration Protocol (DHCP)	B	A				
3.7.3	Internet Protocol version 4 (IPv4)	B	A				
	Internet Protocol version 6 (IPv6)	B	A				
3.8 (U) Serialization	(U) Describe the addressing associated with the following protocols: IPv4 IPv6 Ethernet	B	A				
3.8.1	Demonstrate the ability handle partial reads and writes during serialization and de-serialization	C	B				
3.8.2	Demonstrate the ability to serialize fixed size multi-byte types between systems of differing endianness.	C	B				
3.8.3	Demonstrate the ability to serialize and de-serialize variable sized data structures between systems of differing endianness.	C	B				
3.8.4	Describe libraries commonly used to aid in serialization	B	A				

UNCLASSIFIED//FOR OFFICIAL USE ONLY

3.9 (U) Basic Exam

3.9.1	(U) With references and required resources, pass the provided basic exam to demonstrate proficiency as a basic proficiency CCD.	C	B and C					
-------	---	---	---------	--	--	--	--	--

3.10 (U) Basic Validation Panel

3.10.1	(U) With references and required resources having passed the basic exam, pass the associated validation panel to demonstrate a firm understanding of the work submitted.	C	A					
--------	--	---	---	--	--	--	--	--

(U) APPENDIX A – JQR GLOSSARY

Acronym	Definition
ACI	Army Cyber Institute
ACL	Access Control List
ACS	Army Cyber School
AEP	Advanced Education Program
ANSI	American National Standards Institute
API	Application Programming Interface
ARP	Address Resolution Protocol
AS	Autonomous System
ASCII	American Standard for Information Interchange
ASN	Autonomous System Number
BE	Basic Exam
BGP	Border Gateway Protocol
C2C	Computer To Computer
CDP	Cisco Discovery Protocol
CIDR	Classless Inter-Domain Routing
CLANSIG	Clandestine Signals Intelligence
CMT	Combat Mission Team
CNE	Computer Network Exploitation
CNODP	Computer Network Operations Development Program
COA	Course of Action
CPT	Cyber Protection Team
CRT	C Runtime Library
CSS	Central Security Service
CST	Combat Support Team
DDK	Driver Development Kit
DDS	Digital Defense Services
DHCP	Dynamic Host Configuration Protocol
DISA	Defense Information Systems Agency
DLL	Dynamically Linked Library
DNI	Digital Network Intelligence
DNS	Domain Name System
DOD	Department of Defense
DODIN	Department of Defense Information Network
EECS	Electrical Engineering Computer Science
EEI	Essential Elements of Information
EIGRP	Enhance Interior Gateway Protocol
EWDK	Enterprise Windows Driver Kit
FISA	Foreign Intelligence Surveillance Act
FORNSAT	Foreign Satellite

UNCLASSIFIED//FOR OFFICIAL USE ONLY

FQDN	Fully Qualified Domain Name
FTP	File Transfer Protocol
HTTP	Hyper Text Transfer Protocol
HTTPS	Secure Hyper Text Transfer Protocol
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IN	Information Need
IO	Input / Output
IOS	Internetwork Operating System
IPS	Intrusion Prevention System
IPv4	internet Protocol Version 4
IRP	Integrated Resource Planning
ISP	Internet Service Provider
ISR	Intelligence Surveillance Reconnaissance
JACWC	Joint Advanced Cyber Warfare Course
JCAC	Joint Cyber Analysis Course
JQR	Job Qualification Record
KSA	Knowledge Skills & Abilities
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MAC	Media Access Control
ME	Master Exam
MX	Mail Exchange
NAT	Network Address Translation
NetBIOS	Network Basic Input /Output System
NIST	National Institute of Standards and Technology
NKB	Network Knowledge Base
NMT	National Mission Team
NS	Name Server
NSA	National Security Agency
NSRP	National SIGINT Requirement Process
NST	National Support Team
OPE	Operational Preparation of the Environment
OSPF	Open Shortest Path First
PAA	Protect America Act
PKI	Public Key Infrastructure
POP3	Post Office Protocol
PTR	Pointer
RFI	Request For Information
RIP	Routing Information Protocol
SCI	Sensitive Compartmented Information
SCS	Special Collection Service
SE	Senior Exam

UNCLASSIFIED//FOR OFFICIAL USE ONLY

SI	Sensitive Information
SID	Security Identifier
SIGAD	SIGINT Activity Designator
SIGINT	Signals intelligence
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOA	Start of Authority
SOP	Standard Operating Procedure
SRI	SIGINT Related Information
SSH	Secure Shell
SSO	Special Source Operations
STEM	Science, Technology, Engineering, Mathematics
TACACS	Terminal Access Control Access-Control System
TAO	Tailored Access Operations
TCP	Transmission Control Protocol
TELNET	TCP/IP Network Virtual Terminal Protocol
TFTP	Trivial File Transfer Protocol
TK	Talent Keyhole
TKB	Target Knowledge Base
TL	Team Lead
TRNE	Trainee
TRR	Trainer
TWI	Training With Industry
UAC	User Access Control (UAC)
UDP	User Datagram Protocol
UIS	User Interface Services
USGCB	United States Government Configuration Baseline
USMA	United States Military Academy
USSID	United States Signal Intelligence Directive
UTF	Unicode Transformation Format
UTT	Unified Targeting Tool
VOIP	Voice Over IP
VPN	Virtual Private Network
VSAT	Very Small Aperture Terminal
VTP	Virtual Tunneling Protocol
WAN	Wide Area Network
WAP	Wireless Access Point
WDF	Windows Driver Framework
WDM	Windows Driver Model
WLAN	Wireless Local Area Network

(U) APPENDIX B – REFERENCE MATERIALS

1. C Programming
 - a. The C Programming Language (Kernighan, Ritchie)
 - b. UNIX Man Pages
 - c. C Primer Plus (Prata)
2. Python Programming
 - a. Official Python Documentation
3. Network Programming
 - a. Beej's guide to network programming using Internet sockets (Jorgensen)
 - b. Computer Networking: A Top-Down Approach 7th Edition (James Kurose)
4. Data structures and Algorithms
 - a. Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles Fifth Edition (Narasimah Karumanchi)
5. Design Patterns
 - a. Design Patterns: Elements of Reusable Object-Oriented Software (Gamma, Helm, Johnson, Vlissides)
6. Operating Systems
 - a. Operating Systems: Three Easy Pieces (Arpaci-Dusseau)
7. Secure Coding
 - a. Secure Coding in C and C++ 2nd Edition (Robert C. Seacord)

(U) APPENDIX C – CYBERSPACE CAPABILITIES DEVELOPER (CCD) KNOWLEDGE MATRIX

	Debugging	Testing	Memory	Datatypes	Arrays	Mathematical//Bitwise	File IO	Functions	Data validation	Pointers	Conditional Statements	Network Programming	Compilation	Linking	Virtualization	Concurrency	Persistence	Design patterns	Data Traversal	Secure Programming
C Programming	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	
Python Programming	●	●	●	●	●	●	●	●	●	●	●	●			●	●	●	●	●	
Data Structures	●		●	●				●		●								●		
Algorithms			●		●			●												
Operating System Concepts							●								●	●	●	●		
Secure Coding		●	●						●	●									●	
Networking Fundamentals			●				●		●		●									

UNCLASSIFIED//FOR OFFICIAL USE ONLY

(U) APPENDIX D – MODULE SIGNATURE PAGE

Module 1 – Administrative Module Signatures

NAME: _____ SIGNATURE: _____
TL NAME: _____ TL SIGNATURE: _____
START DATE: _____ COMPLETION DATE: _____

Module 2 – Organizational Primer Signatures

NAME: _____ SIGNATURE: _____
TL NAME: _____ TL SIGNATURE: _____
START DATE: _____ COMPLETION DATE: _____

Module 3 – Individual Competencies Signatures

NAME: _____ SIGNATURE: _____
TL NAME: _____ TL SIGNATURE: _____
START DATE: _____ COMPLETION DATE: _____

