Petition for Academic Credit for Work/Life Experience

Submitted By: [Redacted]
Table of Contents

1. Petition for Academic Credit for Work/Life Experience.................................1
2. Cover letter requesting the LLL credits..........................................................2
3. Resume.............................................................................................................3
4. Narrative that provides detailed experiences related to specific courses....4
   a. justification and documentation of work/life experience.......................5
   b. Example 2......................................................................................................6
   c. Example 3......................................................................................................7
   d. Etc..................................................................................................................8
5. Curriculum Mapping Document........................................................................9
6. Course Syllabi..................................................................................................10
Petition for Academic Credit for Work/Life Experience

Introduction

Requirement: This part of the portfolio sets forth the purpose and answers the who, what, when, where, why, and how relating to the request for academic credit based on life experience.

The purpose of this petition is to request consideration for my life experiences outside of academia. I have a great deal of experience that is directly related to courses I would otherwise need to complete in pursuit of a Bachelor's Degree in Information Technology. Much of this experience has been gained in the dozen years I have spent working in Engineering IT.

Engineering IT is a specialty field that lies at the intersection of Engineering and Information Technology. As an Engineering IT professional, I am responsible for administering and maintaining specialty engineering applications including purchased MCAD design tools and the PDM databases used to store the created content. I also have responsibility for custom application programming, database development, and web development. Additionally, I share responsibility for the hardware platforms and basic operating environments used by our entry and mid-range servers and desktop and mobile workstations.

In 1997, shortly after graduating from high school, I started at [Redacted] in the CAD program. I made the difficult choice between CAD and IT, knowing that the CAD program would keep me close to the IT field due its dependence on computers. I started my career in the fall of 1998 with [Redacted] as a drafter. In January 1999, I received my Associate's Degree in Computer Aided Drafting and Design. I worked in the engineering field as a drafter for about two years. It was during this time that I was first exposed to Unix. The design workstations at our location were all Sun Unix at the time. I picked up enough skills with the Unix OS and a new
3D MCAD package that I was selected for an open CAD support position in the summer of 2000. This position marked my transition from Engineering to Engineering IT.

In my various roles in Engineering IT, I have performed many IT tasks. I started by deepening my understand of the Sun Unix OS, learning many commands and basic shell scripting. I have written and edited Bourne shell, C shell, and Korn shell scripts, as well as scripts written in TCL, Perl, and PHP. I was responsible for benchmarking and selecting workstations. I was instrumental in our switch from Unix workstations to PC based workstations in the early 2000s, completing benchmarking and assessments with workstation platforms from multiple PC and Unix vendors.

Early in my career I learned HTML in order to assist with maintenance of our internal websites. I also learned some JavaScript in order to provide an enhanced web experience. My web skills expanded further into PHP, which is used for a majority of our internal dynamic sites. I used PHP to write a custom extension for Mediawiki that added security functionality necessary for our internal use. I wrote the custom extension in order to make Mediawiki (the software behind Wikipedia) a valid documentation platform for our group.

With basic web and shell programming in hand, I moved on to application programming. My first programs were console based automation programs written in C/C++ using a vendor API to interact with a PDM application. I then experimented with GUI programming using C++ with MFC to write an administrative automation program to interact with the same PDM application. I then moved on to Java, writing a fully automated document release program. The program interacted with the PDM application, the MCAD application, and our internal engineering change database.
When we purchased a product from another company in 2005, I was tasked with updating the models and storing them in a PDM system. I wrote an application in VB.NET that retrieved each model in the MCAD application, updated it per our company standards, and stored it in a new PDM system. This application allowed us to quickly deploy the product in our factories and begin working on the design without requiring a large commitment of man hours to retool per our requirements.

One of my most recent projects was creating and maintaining an automation program to produce custom 3D models for our customers. During the proof-of-concept I developed two automation engines for 3D models, one using a Java API, the other using a C# API. I also assisted another developer who developed a third engine using a VB.NET API. After the proof-of-concept was deemed successful, I developed the database architecture. In subsequent revisions, I have assisted with troubleshooting the server-side PHP code and architected the web-based interface for data entry.

One of my earliest responsibilities was administration of the MCAD and PDM tools. I developed methods to network load the MCAD applications. This is common in Unix, but uncommon in the Windows world. I developed batch scripts to support running the MCAD and PDM client applications that are in use in a dozen facilities around the globe by over three hundred users. At one point these scripts were rewritten in Java, but due to changes in Java system execution and operating systems, they were rewritten back in batch. I have recently supervised a rewrite by a junior programmer in C#, although environmental concerns have kept this from being rolled out.
Administering PDM applications required me to gain knowledge and experience in database systems. Our main PDM applications are built on top of Oracle databases. I have familiarity with Oracle and SQL for various tasks required to install, maintain, upgrade, and otherwise administer Oracle databases. I have also written PHP scripts to interact with the Oracle databases and provide custom reports. Most of our custom internal databases are based on MySQL, so I am also familiar with this database application.

Being responsible for so many server based applications, I have a long record with servers themselves. I assist our Unix server administrator with hardware selection. While he handles most of the Unix installations, I have performed several, including physical Unix servers and workstations and virtual Unix and Linux servers, all from scratch. We adopted VMware many years ago. I have been responsible for the setup, configuration, and maintenance of most of our virtual Windows servers.

As you can see, I have experience from Unix to Windows, desktops to servers, scripting to programming, web development to databases. I also have a deep knowledge of MS Office tools and serve as a local expert for these tools in my Engineering IT team. My experience and skills have served me well throughout my career. When [redacted] purchased [redacted] in 2005, I realized that the new corporate culture placed significantly more importance on formal education than the old. This led to my decision to enroll at Central Penn in [redacted] seeking a bachelor’s degree.

Sincerely,
Engineering IT Professional

Engineering Information Technology professional whose qualifications include multiple disciplines within the field. Extremely fast learner, self-starter, able to work independently or as part of a team. Natural leader with experience leading diverse teams and mission critical projects.

Technical Skill Summary

- **CAD Applications:** Pro/ENGINEER Wildfire 4, SolidWorks 2012, Autodesk Inventor 2012, CADDSS
- **PDM Applications:** Pro/INTRALINK 3.4, Windchill PDMLink 10.0, SolidWorks Workgroup PDM 2012
- **Databases:** Oracle 10g, MySql 5
- **Programming Languages:** Java; C/C++; C#; VBA; Scripting (Windows, Unix); HTML; JavaScript; PHP
- **APIs:** Pro/ENGINEER (J-Link, Pro/TOOLKIT, Pro/INTRALINK Toolkit; Autodesk Inventor; SolidWorks (SolidWorks, Workgroup PDM); Java (JavaMail, JDBC)
- **Systems:** Windows 7 (32 and 64 bit); Unix (Solaris 10); VMware
- **Other Software:** MS Office 2010 (Excel, Word, PowerPoint), MS Project, MS Visio, CA Clarity

Experience

Staff Information Technologist Project Manager
Principal Engineer / Technologist
Senior Engineer / Technologist
Software Engineer III
Senior CAD Support Specialist
CAD Support Specialist

- Support key initiatives. Completely support CAD and CAM. Assist in selection of workstation and server hardware including benchmarking CAD applications. Manage and deploy server environments and applications. Investigate and implement new technologies to better the Engineering environment. Develop software applications to automate tasks and provide services to internal and external customers. Articulate and optimize new technologies. Manage software versions and upgrade projects. Adhere to and administer information security requirements for commercial and government engineering groups.

Notable projects include:
- Provided technology and business process expertise while serving as key contributor on the business unit PLM application selection core team.
- Developed scripts to enable network loading and remote synchronization of PTC software to remote facilities to create a standardized global platform for over three hundred users in more than ten global locations.
- Developed an application that automated release drawing file creation including verifying information from the EC database.
- Developed automation which updated a purchased product to company standards and imported into PDM system, minimizing the expense and time-to-market for the acquired product.
- Implemented a wiki for documentation.
- Architected and developed automated BIM model generation system for highly configurable products. Served as project lead for BIM team including development, maintenance and data entry.
- Architected and managed the development of a web-based database application for an internal engineering team to facilitate product globalization.

- Produced quality models and drawings for manufacturing at the local plant using CADDSS and Pro/ENGINEER while meeting schedule requirements. Automated part creation using Pro/Program.

Key Roles and Responsibilities

- **Leadership**
- **Mentoring**
- **Team Development**
- **CAD Installation, Configuration, and Administration**
- **PDM Installation, Configuration, and Administration**
- **Software Architect and Developer**
- **Innovator**
- **Database Installation, Administration, and Architecture**
- **Project Management**
- **Technical Leadership**
- **Contract Management**
- **Technology Adopter**
- **User and Server Hardware Selection**

Education

Central Pennsylvania College, Summerdale PA
Bachelor’s Degree in Information Technology

York Technical Institute, York PA
Associates Degree in Computer Aided Drafting And Design
Content and Description

Requirement: This part of the portfolio must provide a detailed description of each course for which credit is requested. It is an embellishment of the short description found in the college catalog and must outline the nature of the course in terms of concepts, theories, models, functions, and procedures.

In this section I have included the relevant courses, their descriptions, and their associated objectives. The descriptions and objectives were extracted directly from the college’s syllabus for each course.

1. TEC235 Object-Oriented Programming I

This course is an introduction to programming and algorithm development using an object-oriented language (currently C++). Topics covered are basic data types, input, output, control structures (selection, repetition), user-defined functions, arrays, strings, and records.

KNOWLEDGE OBJECTIVES

1. List the steps required to execute programs in a high-level language.
2. List and describe the basic operations of a computer.
3. List and describe the attributes of a program variable.
4. Name and describe the process of utilizing external files for program input or output.
5. Explain sequencing, selection, and repetition.
6. Distinguish when one control structure is “better” to use than another.
7. Distinguish between pre-defined and user defined functions.
8. Explain the difference between passing parameters by value and by reference.
9. Determine when to use pass by value and pass by reference.
10. Distinguish between homogeneous and heterogeneous structured data types.
11. Describe the steps in designing an object-oriented computer program that will meet problem requirements.
12. Describe the steps one takes to debug a program so that it will execute successfully.

PERFORMANCE OBJECTIVES

1. Compile, link, and execute a program written in a high level language.
2. Write programs utilizing variables, assignment, input and output.
3. Write programs using input/output (I/O) operators and manipulators.
4. Write programs using standard I/O and file I/O.
5. Use sequencing, selection, and repetition in writing programs.
6. Write programs using pre-defined functions.
7. Construct user-defined functions.
8. Write programs utilizing enumerated types, string types, record types, and vector types.

2. TEC242 Linux

This course provides a structured approach to attaining the theoretical knowledge and practical skills to installing and configuring a Linux/Unix operating system. The course includes detailed information on how Linux/Unix addresses operating system issues, such as the kernel, file and directory structure, shells, shell programming, file processing, and the X Windows system. Upon successful completion of this course, students will have learned the requisite knowledge to install, configure, and use a Linux/Unix operating system.

KNOWLEDGE OBJECTIVES

1. Describe the characteristics of the Linux/Unix operating system.
2. Explain the architectural differences between Windows and Linux/Unix systems
3. Explain the functions of the kernel, the shell, and applications
4. Describe the Linux/Unix file and directory structure.
5. Explain the relationship between X and the Linux GUI
6. Explain the fundamental differences between different distributions

PERFORMANCE OBJECTIVES

1. Install and configure a Linux OS.
2. Use Linux shells to create and manipulate files.
3. Use vim or other text editor to create and edit files.
4. Write simple shell scripts to automate tasks.
5. Identify, locate, install, and configure programs from source code
6. Identify, locate, install, configure and uninstall packages.
7. Compiling the Linux kernel
8. Configuring X Windows system

3. TEC245 Internet Programming I

This course is an introduction to tools and programming skills needed for developing Internet applications. Topics covered include HTML, scripting languages such as JavaScript, XML, PHP, and database access through the web.
KNOWLEDGE OBJECTIVES

1. Describe the concepts involved with Internet programming.
2. Explain how the components of Internet programming work in concert to afford the programmer source code that is more easily maintained.
3. Compare and contrast HTML and XHTML
4. List and describe the various scripting languages currently in use in business.
5. Distinguish between client-side and server-side programming
6. Explain the programming aspects of accessing a database through the web.
7. Identify and describe the qualities and behaviors necessary for positive business relations and effective communication.

PERFORMANCE OBJECTIVES

1. Illustrate knowledge of the steps to design, develop, execute, and debug a visual, object-oriented, Internet computer program.
2. Demonstrate knowledge of applying visual design techniques and object-oriented concepts to Internet Web page design.
3. Apply knowledge of the correct application of visual components, object-oriented code, and Internet design requirements to answer specific business needs.
4. Demonstrate and apply effective written and oral communication.
5. Demonstrate the ability to complete science and math problems or case studies.

4. TEC252 SQL Fundamentals

Structured Query Language is the standard language for creating, maintaining, and extracting data from all relational databases. SQL Fundamentals introduces the conceptual framework for SQL and syntax for identifying and defining data types, defining the relationship between fields, and entering data into the proper table. Students will be expected to learn how to enforce data integrity and query a database within SQL.

KNOWLEDGE OBJECTIVES

1. Understand the unique structure and components that are characteristic of SQL.
2. Discuss syntactical requirements for using SQL with a relational database.
3. Describe the different data types and how they are used in a relational database.
4. Articulate how to construct tables, enter and modify data, create views, and extract data using SQL.

PERFORMANCE OBJECTIVES
1. Demonstrate knowledge of the structure and syntax of standard SQL.
2. Define data types, field relationships and constraints to construct a table using standard SQL syntax.
3. Enter data in a standard table and modify table structure using standard SQL syntax.
4. Demonstrate the ability to access data using standard SQL syntax and functions.

5. TEC262 Fundamental Database Administration

   With a working knowledge of standard SQL, students will be introduced to the principles of database administration, including managing user access, populating tables, and modifying tables. Students will learn to install and start a database server.

KNOWLEDGE OBJECTIVES

1. Outline the method and procedure for performing common DBA tasks including user access management.
2. Describe the method and procedure for creating and populating relational database tables.
3. Explain the methods and procedures for altering table definitions and the impact those changes will have on database integrity.
4. Describe the method and procedure for performing backup and restore operations.
5. Explain locks and transaction processing.
6. Analyze database requirements and use proper naming conventions to create appropriate data elements and data types.
7. Demonstrate knowledge of techniques to instruct a database to process data and output results.

PERFORMANCE OBJECTIVES

1. Identify, download, and install the appropriate open source database software.
2. Analyze database requirements and use proper naming conventions to create appropriate data elements and data types.
3. Correlate data elements to create and relate tables in the database implementation.
4. Demonstrate knowledge of techniques to modify tables and insert and delete data.
5. Demonstrate knowledge of techniques to instruct a database to process data and output results.
6. Demonstrate knowledge of techniques of basic user access management.
7. Integrate effective written and oral communication into daily business practices.
8. Generate results by utilizing science or mathematical skills and apply those results to problem solving.

6. TEC335 Object-Oriented Programming II
This course is a continuation of Object-Oriented Programming I. Topics covered include classes of inheritance, polymorphism, and basic abstract data types including lists, stacks, queues, debugging techniques, and program compilation and execution. An object-oriented programming language (currently C++) will be used to demonstrate constructs and syntax.

KNOWLEDGE OBJECTIVES

1. Understand the concepts involved with Object-Oriented Programming (OOP) and its implementation in C++.
2. Describe the steps in designing an object-oriented computer program that will meet problem requirements.
3. Explain the steps one takes to debug a program so that it will execute successfully.
4. Understand how the components of OOP work in concert to afford the programmer source code that is more easily maintained and reused.
5. Compare the relative efficiency of various implementations of an abstract data type in terms of Big-Oh notation.
6. Explain information hiding and encapsulation.
7. Trace the execution of a program using pointers.

PERFORMANCE OBJECTIVES

1. Write programs demonstrating knowledge of the steps to design, develop, execute, and debug a visual object-oriented computer program.
2. Implement a variety of abstract data types including list, stack, and queue.
3. Write programs utilizing linked structures (pointers)
4. Integrate effective written and oral communication by documenting code and presenting solutions.
5. Generate results by utilizing science or mathematical skills and apply those results to problem solving.
Justification

Requirement: This part of the portfolio must show how the student's work/life experience relates explicitly to the course. Students must talk the language of the course and connect the discussion with appropriate enclosures that demonstrate the student gained the theoretical knowledge of the course by a combination of theory and practice. Diplomas, certificates, work projects, awards, letters from immediate supervisors, and other accomplishments are appropriate enclosures to substantiate the justification discussed in this part of the request.

Course Specific Justification

1. TEC235 Object-Oriented Programming I

I have experience with Object-Oriented Programming in multiple languages including C++, Java, C#, and PHP. I have developed C/C++ applications in Unix using make files and compiling and linking on the command line. I have also used Visual Studio on Windows to develop C/C++ and C# applications. I have experience debugging in each of these languages using the tools available. I prefer to use an IDE such as NetBeans or Visual Studio for their enhanced debugging capabilities. It is extremely beneficial to follow the execution of code, watching the values of variables change to understand program operation and troubleshoot problems. I especially like the remote debugging capabilities included with Java, which allow me to debug issues in running code on my near-production virtual server instead of having to try to replicate a server environment on my development PC.

I am familiar with programming constructs and flow control mechanisms. I have developed programs from scratch that are currently in production for my employer. I am comfortable writing various types of objects including classes, interfaces, and enums. I am also familiar with using abstraction to implement hierarchical class design. I have used many types of variables including signed and unsigned basic number types (int, short, long, double, float) in C/C++ and more advanced Strings in Java. I understand the difference between pass by value, where a local copy of the variable is made at runtime to prevent modification of the original
variable, and pass by reference, where the memory location of the variable’s value is passed into the function/method and modification of the original is possible. My first introduction to this concept was learning to employ pointers and references (&) in C/C++.

I have implemented I/O using both command lines and files. Most of the work I have done takes advantage of file I/O, reading a source file, transforming it, operating on it, and producing the necessary output. One such program I wrote recently takes a CSV file as input and produces a properly formatted output file for each line the CSV file contains.

Included are examples of basic OOP written by me from scratch for my employer. These examples serve to demonstrate my grasp of OOP concepts and skills. All code is property of Johnson Controls.

2. TEC242 Linux

I have used the Sun Solaris operating system since 1998 at work on a regular basis. I started on a Sparc 20 workstation running a CAD application. I quickly learned Vi, although most of the time we use Vim now. Up until the recent batch of Windows programming IDEs, I would frequently take files to Unix for editing because the substitution and global replace capabilities built into Vi are far superior to anything notepad offers. Now that NetBeans and Visual Studio include regular expression search and replace, this is no longer an issue. I still return to Vi often for its advantages in opening large files that Windows editors simply cannot process.

I have used different shells in my time as a Unix user and administrator including Bourne, C shell, and Korn shell. I have written and edited many shell scripts to perform various tasks. While our Unix administrator does most of the work installing servers, operating systems, and applications, I have performed these tasks myself when the need has arisen. I installed our
first instance of Solaris x86 on a VMware platform. I am comfortable using configure and make
to compile local applications. This is how our system administrator prefers to deploy
applications so that we have as much control as possible over the what, where, and how of
installation. I have also installed software using packages in Solaris, notably Java. I also have an
Ubuntu thumbdrive that I built and used for a TEC222. I used the package features in Ubuntu
Linux to install updated versions of Firefox and Nmap for use in this course.

We use the X Windows program Xming at my facility to remotely display sessions from
our Unix servers. I have installed and configured Xming, and I maintain the configurations that
are used by all users including the X Windows options necessary for proper functionality.

Included are sample shell scripts I have worked on. All code is property of Johnson
Controls. Also included is a recommendation from our Unix System administrator.

3. TEC245 Internet Programming I

I learned HTML in 2000 so that I could assist with our department’s intranet website.
We actually host and maintain multiple websites on multiple virtual Unix servers running
Apache. I prefer to write and edit HTML by hand, although some of the tools in newer IDEs are
useful at times. I took over one of our websites in the early 2000s and wrote the entire site by
hand myself, including the HTML, CSS, supporting JavaScript, and content.

I have also written and edited PHP pages. I deployed MediaWiki in our environment
several years ago. For our use, one of the main shortcomings in the MediaWiki platform is its
lack of security. I researched the extensions that were available at the time to add security
features, but I found a bug in the best one that showed search results for secure pages to
anonymous users. I wrote a separate extension to address this issue in object-oriented PHP.
Later when the maintainers of MediaWiki took a harder stance against security extensions, I wrote a custom extension that allows us to run multiple MediaWiki instances using a custom authentication to a MySQL database so that only one set of standardized credentials is necessary.

Included are samples of Web content showing HTML, CSS, and JavaScript written by me from scratch for my employer. Also included is a PHP script that accesses an Oracle database and creates HTML output to detail the security access configurations of a PDM system. All code is property of [Redacted].

4. TEC252 SQL Fundamentals

For over ten years I have been a system administrator for a purchased database application that is built on top of Oracle. As the administrator for this database, I have spent a lot of time interacting with the database via SQL. I am familiar with common SQL commands and statements. In addition to Oracle, we also use MySQL for most of the custom in-house databases in my group. I have used tools from the command line to phpMyAdmin to administer and interact with various databases. I have a firm grasp of database design, including constraints. I have served as database architect for several in-house projects. I can write SQL scripts to perform interaction with a database, for querying data as well as for modifying data.

Included are samples of SQL scripts written by me for my employer. All code is property of [Redacted].

5. TEC262 Fundamental Database Administration

As described for TEC252, I have a long period of experience as the system administrator for my group's Oracle databases. I also have experience with MySQL databases. I have served as
the database architect for several internal projects, including a production application that makes customer models and an internal web-based application for an engineering group. As the architect, I designed the data structures and table layouts, normalized the tables, and oversaw the implementation. As the administrator for these database applications, I generate custom output and input new and revised data as necessary to assist in the overall maintenance of the system. The custom work I do requires knowledge of SQL and writing queries to process complex situations or large quantities of data.

6. TEC335 Object-Oriented Programming II

As described for TEC235, I have many years of experience with Object-Oriented Programming with various languages including C/C++, C#, Java, VB.NET, and PHP. I have written production applications from scratch, including all of the testing and debugging necessary to make a properly functioning, well tested application. I have used pointers in C/C++, although I do prefer languages like Java and C# that omit pointers.

Included are samples of OOP code written by me for my employer. All code is property of
Letters of Recommendation

Included are three letters of recommendation from current colleagues. I have worked with each of these individuals for more than a decade. Details about these individuals are:

- [Name] - System Administrator for Unix and Windows servers, web developer
- [Name] - My direct supervisor, Manager for [Company]
- [Name] - Director of Global [Company]

In addition to the letter, [Name] has offered to speak personally about my petition. He can be reached at [Contact Information]
Letters of reference contain sensitive information and cannot be provided.
Work Examples

Enclosed are samples of my work as indicated in the course specific justification sections above. The specifics of the enclosures are as follows:

- CProRelease.cpp – (33 pages) A C++ source file that is part of a large program. The C++ was used to interact directly with a vendor supplied C/C++ API. The majority of the program was written in Java, with the C++ code accessed via JNI. This code provides several examples of C++ flow control, custom variables and functions, pointer use, and I/O. Note: several strings were redacted from this example.

- Form1.cs – (3 pages) A C# source file that is part of a small application. This application was used to test an RSS style output implementation for an automation engine with a web front end. This code provides several examples of C# flow control, custom variables, and custom functions.

- ilink_permission_report.php – (5 pages) A PHP source file that is executed from the command line as a utility. This utility queries an Oracle database and uses the results to write an HTML formatted report. This code provides examples of PHP flow control, custom variables, I/O, HTML and SQL.

- ilink_user_permission_report.php – (4 pages) A PHP source file that is executed from the command line as a utility. This utility queries an Oracle database and uses the results to write an HTML formatted report. This code provides examples of PHP flow control, custom variables, I/O, HTML, and SQL.
- GetFilesFromObjects.java – (10 pages) A Java source file that is part of a small utility. This utility reads a large report file, building logical objects from the contents, and finds matching data from a separate input file for further examination. This code provides many examples of Java flow control, custom variables, custom functions, custom objects, and I/O.

- yiptcsiweb.css – (2 pages) A cascading style sheet (CSS) file used to provide consistent formatting for an intranet website.

- yiptcsi.js – (2 pages) A JavaScript source file used with an intranet website. The functions provide special handling for frames, framesets, and the base window to ensure the website is displayed properly. This code provides examples of JavaScript, flow control, custom variables, and custom functions.

- wmpobj.js – (2 pages) A JavaScript source file used with an intranet website. The functions provide support for embedding a Windows Media Player object to play videos with a custom codec. This code provides examples of JavaScript, HTML, and custom functions.

- skelcn.html – (1 page) An HTML source file describing a procedure from an intranet website. This file shows basic HTML elements including headings, paragraphs, and lists, as well as script elements to display a video.

- sysarch.html – (1 page) An HTML source file describing an application structure from an intranet website. This file shows basic HTML elements including headings, paragraphs, lists, and anchors, as well as use of CSS elements.
- custom.html – (2 pages) An HTML source file describing an application structure from an intranet website. This file shows basic HTML elements including headings, paragraphs, and lists, as well as use of CSS elements.
- .cshrc – (2 pages) A Unix C shell startup file containing the setup for a database user. This file shows understanding of Unix setup and commands.
- findflex.csh – (1 page) A simple Unix C shell script used to list the contents of all tar files in a directory. This file shows understanding of Unix commands and scripting.
- runme.ksh – (1 page) A simple Unix Korn shell script used to manipulate a large quantity of output files from an application and post-process the data. This file shows understanding of Unix commands and scripting.
- update_fs.sql – (1 page) A simple SQL script that runs another script and then takes input and updates records.
- email_list.sql – (1 page) An SQL script that lists the email addresses of users in a database. The users are only listed if they do not match a qualifier.
- usr_loc_rpt.sql – (1 page) An SQL script that lists users in a database with specific attributes and displays them grouped by a specified attribute.
- dbf_rpt.sql – (1 page) An SQL script that reports the space used by database tables in human readable form.
- dbf_warn_rpt.sql – (1 page) An SQL script that lists any database tables that contain less than a given percentage of free space.
Work examples contain sensitive information owned by a third-party employer and cannot be provided.