FLORIDA INTERNATIONAL UNIVERSITY
UNIVERSITY CURRICULUM COMMITTEE
Proposal for a New Course

Bulletin # : _____

Academic Year : _____

1. **School/College** College of Engineering and Computing

   Div./Dept. in Which Taught Knight Foundation School of Computing and Information Sciences

2. | Alpha Prefix | 1st Digit | Last 3 Digits | "C"-lec-lab "L"-Lab | Cr. Hrs. |
   |---|---|---|---|---|
   | CIS | 3 | | | 0 |

   CIP Code ***(Leave this blank):*** _____

3. **Grading Method (select one):** ☒ Graded ☐ Pass/Fail

4a. **Course Title** Internship Ready Software Development

b. **Abbreviated course Title** ***(for computer class schedules, transcripts)*** Internship Ready SWDev

LIMITED TO 25 Characters *(including spaces)*

5. **Statewide Course Numbering Subject Matter Area** Computer Science and Information Systems

6. **Catalog Description/Major Topics** *(not to exceed 200 characters including spaces)*

College of Medicine and College of Law: Attach description not exceeding 1,000 characters including spaces.

Gain effective hard and soft skills for a successful software engineering internship. Topics covered include software version control, project management, open-source development, and AI.

7. **Attach detailed syllabus course outline and course justification on separate page(s).**
8. **Prerequisite(s):** COP 2210 or COP 2250 or equivalent
9. **Corequisite(s):** COP 3337 or COP 3804 or equivalent
10. **Objective(s) of Course:**

Apply the best practices for effective project management techniques, version control for software development, and evaluate the advantages and challenges of open-source software development.

11. **Does this course duplicate/overlap other courses at FIU?** ☒ No ☐ Yes
    If yes, please explain: _____
12. **What other closely related department(s) have been consulted about this course?**

    _____
13. **Is this course used for the assessment of a program or a certificate (if yes, then send a notification to assessment@fiu.edu)?** ☑ **No** ☐ **Yes**

PROPOSAL REQUESTED BY:

Faculty Contact Juan Mancilla-Caceres (Type name) _____ (Signature) 12 / 19 / 2023

jmancill@fiu.edu (Email address) 2177218610 (Phone number)

Chairperson (Dept./Div.) Jason Liu (Type name) _____ (Signature) ____ / ____ / 20____

Chairperson (Curr. Comm.) Alexander Afanasyev (Type name) _____ (Signature) ____ / ____ / 20____

College/School Dean Ines Triay (Type name) _____ (Signature) ____ / ____ / 20____

Submit one original form. Attach one copy of the course justification and a draft of the course syllabus for this New Course Proposal. *The complete syllabus should include all components listed on the New Course Checklist.*

Internship Ready Software Development - Justification

This course will provide effective skill training for students to have a successful Software Engineering Internship. It will allow the student to gain experience and confidence using version control, agile project development, and developing open-source software in an active learning environment within nine weeks. The need for this comes from conversations with industry partners and a commitment from KFSCIS to provide tech career awareness, readiness, upskilling, and other wraparound services tailored to improve tech students' employment success and alumni & company engagement in career development activities. Feedback will be given to students in a managerial style, e.g., through code reviews and sprint retrospectives.

The course prepares students for their first internship and post-graduation employment by bridging knowledge gaps, building software development portfolios, strengthening soft skills, and increasing confidence. Specifically, the instructor and learning assistant will guide computing students in building open-source software portfolios, which are critical to impress recruiters and hiring managers and are considered an industry best practice for acquiring a software development job.

According to the Linux Foundation's 2022 Open Source Jobs Report, 93 percent of employers have difficulty hiring open source talent. Growth in this area is projected to accelerate to at least 18.2 percent through 2026. Fig. 1 shows the growing importance of open source for employment in computer science.

Students will start and/or contribute to an Open Source software project as a team by (1) using Git and Github, (2) applying Scrum software project management techniques, (3) learning a core framework or platform such as artificial intelligence application programming interfaces as part of their project. Learning will be self-paced, peer group organized, learning assistant guided, and appropriate for their skill level —building a career development community support system so students can support each other. In Fig. 2, from the above-mentioned report, the hiring demand and perceived need for tech talent with open source skills are highlighted. The survey conducted in March 2022 among 1,672 open-source professionals and 559 respondents responsible for hiring open-source professionals shows the significance of open-source skills in the tech industry.

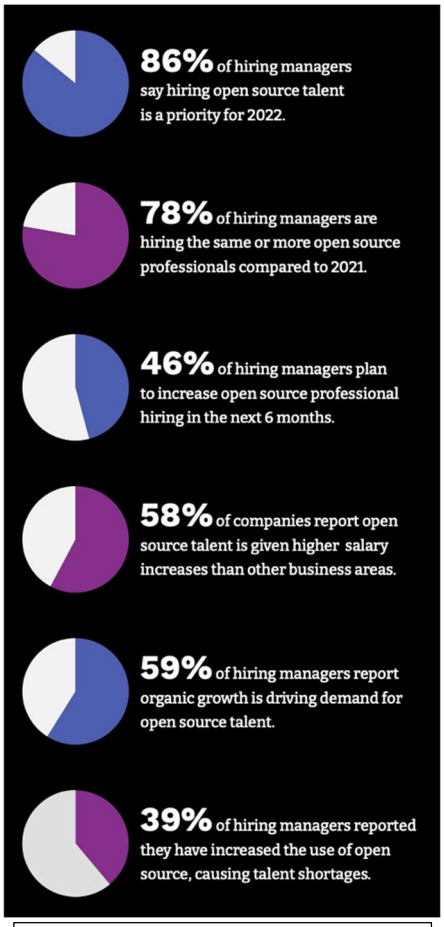

Fig.1 Open Source Jobs Report

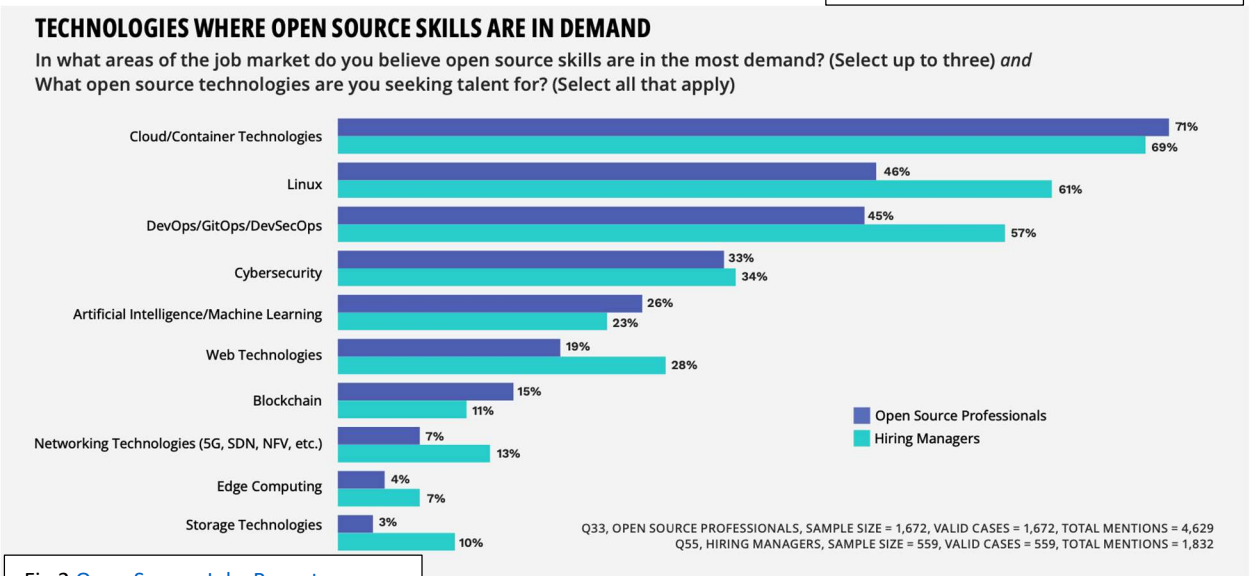

Fig.2 Open Source Jobs Report

Knight Foundation School of Computing and Information Sciences

Course Title: Internship Ready: Software Development **Date:**

Course Number: CXX 3XXX

Number of Credits: 0

Subject Area: XX	**Subject Area Coordinator:** XX **email:** XX@fiu.edu
Catalog Description: This course provides effective skill training for students to have a successful software engineering (SWE) Internship. It will allow the student to gain experience and confidence using version control, agile project development, and developing open-source software in an active learning environment within a 9-week period.	
Textbooks: **TBD**	
References: **TBD**	
Prerequisites Courses: COP 2210 Computer Programming I	
Corequisite Courses: COP 3337 Computer Programming II	

Type: Zero Credit Elective for CS, IT and Cybersecurity Data Science Majors

Prerequisites Topics:

1. Fundamental programming
    a. Objects, Classes, and fundamental Data Types
    b. Arrays, Strings, and Functions
2. Taking a course with Data Structures as co-requisite
3. Oral and written communication skills

Course Outcomes:

1. Apply the best practices for effective project management techniques.
2. Analyze various strategies and tools for developing a successful software development project.
3. Apply version control for software development
4. Examine the importance of version control in maintaining code.

5. Evaluate the advantages and challenges of open-source software development.

Relationship between Course Outcomes and Program Outcomes

Student Learning Outcomes	Course Outcomes
1. Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions.	1,2,3
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.	1,2,5
3. Communicate effectively in a variety of professional contexts.	1,3,4
4. Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	2,5
5. Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.	1,3,5

Knight Foundation School of Computing and Information Sciences
C **** Internship Ready: Software Development**

Assessment Plan for the Course and how Data in the Course are used to assess Student Outcomes

Outline

Topic	No. of Lecture Hours	Course Outcomes
Version control with Git and GitHub		
• Describe why version control is a fundamental tool for coding and collaboration • Install and run Git on your local machine • Perform common tasks in GitHub. • Collaborate with others through remote repositories • Command-Line Interface / Integrations with various IDEs Assessment: • Managerial-style review of policies and feedback: o Was a repository created? o Does it follow guidelines? o Is it secure? o Is it properly described? o Are the commits properly described?	5	1,2,3
Open-Source Development		
• Define what open-source software is. • Identify the benefits OSS has provided to the world's technology infrastructure. • Describe collaboration best practices. • Encourage skill diversity in teams working in open-source projects. • Describe common methods, such as Continuous Integration, and use of GitHub and other hosting providers. Assessment: • Managerial-style review and progress report: o Were relevant OSS properly identified and research? o Was a proper licensing selected? o Were the policies of the selected OSS reviewed? o What are the challenges and opportunities?	5	1,2,4
Project Management		
• Adopt the 5 practices of Agile, a subset of DevOps: small batches, minimum viable product, pair programming, behavior- and test-driven development. • Write good user stories, estimate, and assign story points and track stories using a kanban board. Incorporate Scrum artifacts, events, and benefits. • Create and refine a product backlog using the sprint planning process. Produce potentially shippable product increments with every iteration.	10	1,2

• Create burndown charts to forecast the ability to meet a sprint goal. Use metrics to enhance performance, productivity, and client satisfaction. • Describe the five important Scrum events and how to set up each event for a Scrum team. Assessment: • Managerial-style product review: o Review of User Stories o Were the sprints and daily stand ups set appropriately? o Are they using appropriate tools?		
Special Topics Reviews		
• Learn the general concepts of data visualization along with basic methodologies and applications. • Implement Dashboards using open-source and proprietary tools. • Identify the best patterns for data visualization of numerical and non-numerical data. • Practice best strategies for communication of results. • Give a pitch Assessment: • Managerial-style review: o Feedback on presentation o Was the audience properly identified and addressed? o Was there any area for improvement?	2	1,2
• Recognize the opportunities and challenges of the applications of Artificial Intelligence in software development. • Evaluate Machine Learning Models. • Using application programming interfaces (APIs) to transform data. • Identify the best practices of safe use of AI. Assessment: • Technology review: o Are all the risks and opportunities identified? o Are API's used effectively? o Is there a plan for testing and deployment?	3	1,2
Guided Development		
• Execute the proposed plan of a project. • Implement the best practices of project management in a real case scenario. • Perform version control of a project using GitHub. • Manage the scope of your project based on the current timeline. Assessment: • Code review: o Does the code follow guidelines? o Is it properly documented? o Has it been properly tested?	15	1,2

• Perform a product presentation to a general audience.
• Implement the best practices of technical communication
Assessment:
• Start-up-style pitch review:
  o Feedback on presentation
  o Was the audience properly identified and addressed?
  o Did the solution address all the requirements?
	5	1

Oral and Written Communication:

Final presentation may include poster, presentation and / or written report.

Social and Ethical Implications of Computing Topics:

Proper use of Open-Source Licenses

Theoretical Contents:

No Significant Coverage

Problem Analysis Experiences:

No Significant Coverage

Assignment 1: Use Scrum / Agile to Manage your Project.

Student Learning Outcome 1: Apply the best practices for effective project management techniques.

Objective: The purpose of this assignment is to familiarize you with the typical strategies and techniques of Scrum for managing team projects.

Task: Using the techniques and tools shown in class, set up a sprint and daily stand ups using one of the project management tools available for you. You are expected to create a set of User Stories describing at least 2 features of your project based on the feedback you got from your audience. At the end of the course, you will present your project, deliver documentation about the user stories, and deliver a retrospective document showing the activities of each sprint and daily standup. During your presentation you are expected to identify and describe your audience, the objectives of your project and features and how they respond to the needs of your audience.

Time to complete: This task is to be worked on a weekly basis, and final deliverables (including presentation, and documents) should be submitted by the end of the course (a total of 9 weeks).

Documentation requirements:

- Please follow through with the guidelines given during class.
- There is no minimum or maximum length of the documentation, but it should follow established guidelines (see references below).

Resource:

- How to Write User Stories: https://www.atlassian.com/agile/project-management/user-stories
- How to do Sprint Planning: https://plan.io/blog/sprint-planning/
- How to run and document standups: https://notejoy.com/resources/stand-up-meetings#:~:text=Reduce%20the%20unnecessary%20documentation&text=That%20said%2C%20it%20can%20be,attendees%20to%20briefly%20fill%20out.
-

Passing criteria: 80% (24/30 pts)

Rubric:

SLO 1 Rubric					
Criteria	**Ratings**				**Pts**
Review of User Stories	**5 to > 4 pts** **Excellent** User stories are well documented and completely capture the features of the use case.	**4 to > 2 pts** **Proficient** User stories are well documented but miss some of the important features the user desired.	**2 to > 1 pts** **Limited** User stories are incomplete and lack documentation	**1 to > 0 pts** **Poor** No user stories created	**5**
Sprints and Daily Stand Ups	**5 to > 4 pts** **Excellent** Weekly Sprints and Daily stand ups are conducted at the right times and properly documented.	**4 to > 2 pts** **Proficient** Some Sprints and Stand ups are missing but they are properly documented	**2 to > 1 pts** **Limited** Documentation is missing of either sprints or standups	**1 to > 0 pts** **Poor** No Sprints or Stand Ups defined	**5**
Tools	**5 to > 4 pts** **Excellent** Team set up a project management tool and shows evidence of regular use	**4 to > 2 pts** **Proficient** Tool is set up but is not regularly used	**2 to > 1 pts** **Limited** Tool is set up but never used	**1 to > 0 pts** **Poor** No tool set up	**5**
Presentation	**5 to > 4 pts** **Excellent** Objectives are well explained, slides are sufficient and high quality, presentation did not went over time.	**4 to > 2 pts** **Proficient** Slides were too few or too many, presentation effective but too long or too short.	**2 to > 1 pts** **Limited** Incomplete presentation, objectives not clear, audience not correctly identified	**1 to > 0 pts** **Poor** No presentation	**5**
Identification of audience	**5 to > 4 pts** **Excellent** Objectives perfectly matched to audience	**4 to > 2 pts** **Proficient** Some mismatch between objectives and audience, but presentation still effective	**2 to > 1 pts** **Limited** Objective not clearly matched to audience but evidence of effort is present	**1 to > 0 pts** **Poor** No identification of audience.	**5**

Coverage of topics and features during presentation	**5 to > 4 pts** **Excellent** All features described in user stories were addressed during presentation	**4 to > 2 pts** **Proficient** Some features were missing during the presentation	**2 to > 1 pts** **Limited** User stories not addressed sufficiently during presentation	**1 to > 0 pts** **Poor** Complete mismatch between features and presentation	**5**
				Total Points:	**30**

Assignment 3: Use an API and write code to implement features described in User Stories.

Student Learning Outcome 2: Analyze various strategies and tools for developing a successful software development project.

Objective: The purpose of this assignment is to evaluate your implementation and proper usage of APIs to solve the problem defined in previous assignments.

Task: Following the plan created in Assignment 1 and Assignment 2 (the user stories and open-source projects) you are to implement the features and solve the problems you and your team identified. You are expected to write proper code documentation and to research any library or API that you decide to use. Your code will be reviewed on a weekly basis and by the end of the course it will be evaluated. Your code should include unit tests as explained in class and you should have clearly defined KPIs and/or OKRs as relevant to your project.

Time to complete: This task is to be worked on a weekly basis, and final deliverables (including code and documents) should be submitted by the end of the course (a total of 8 weeks).

Documentation requirements:

- Please follow through with the guidelines given during class.
- There is no minimum or maximum length of the documentation, but it should follow established guidelines (see references below).

Resource:

- How to Document Code: https://dev.to/digvijayjadhav98/code-documentation-a-guide-for-beginners-4cj7
- How to Document APIs: https://idratherbewriting.com/learnapidoc/docendpoints.html
- KPIs and OKRs: https://www.wrike.com/blog/kpis-vs-okrs-compare-need-successful/#:~:text=Objectives%20and%20key%20results%20(OKRs,re%20heading%20the%20right%20way.

Passing criteria: 80% (24/30 pts)

Rubric:

SLO 2 Rubric					
Criteria	**Ratings**				**Pts**
API Risks and Opportunities	**5 to > 4 pts** **Excellent** Complete description of API functionality, strengths and limitations.	**4 to > 2 pts** **Proficient** Partial knowledge of API and cursory review of limitations.	**2 to > 1 pts** **Limited** Little understanding of API	**1 to > 0 pts** **Poor** No analysis of API's capabilities	**5**
Effective use of APIs	**5 to > 4 pts** **Excellent** API used as expected and properly documented.	**4 to > 2 pts** **Proficient** API used without proper documentation.	**2 to > 1 pts** **Limited** Limited use of API	**1 to > 0 pts** **Poor** No API usage or wrong understanding of its functionality	**5**
Code review	**5 to > 4 pts** **Excellent** Code follows the guidelines completely (e.g, size of functions, naming convention, etc.)	**4 to > 2 pts** **Proficient** Some guidelines are not followed but are justified	**2 to > 1 pts** **Limited** Guidelines are rarely followed without proper justification	**1 to > 0 pts** **Poor** Guidelines are not followed at all	**5**
Code Documentation	**5 to > 4 pts** **Excellent** Code properly and fully documented (e.g., description of each function and procedure, using correct markup annotations)	**4 to > 2 pts** **Proficient** Code documentation mostly present but missing in some parts.	**2 to > 1 pts** **Limited** Documentation missing or incomplete	**1 to > 0 pts** **Poor** No documentation	**5**
Code Testing	**5 to > 4 pts** **Excellent** Unit tests for all relevant code implemented and properly executed.	**4 to > 2 pts** **Proficient** Some tests implemented but not covering all cases	**2 to > 1 pts** **Limited** Tests present but not correctly implemented	**1 to > 0 pts** **Poor** No testing.	**5**

Code addresses requirements	**5 to > 4 pts** **Excellent** KPIs and OKR properly defined and evaluated	**4 to > 2 pts** **Proficient** KPIs and OKR identified but not improved	**2 to > 1 pts** **Limited** KPIs not properly defined	**1 to > 0 pts** **Poor** No KPIs identified	**5**
				Total Points:	**30**

Assignment 4: Apply Version Control Effectively

Student Learning Outcome 3: Apply version control for software development and its importance in maintaining code.

Objective: The purpose of this assignment is to apply your knowledge of version control in an effective way to manage your own and your team's code.

Task: In this task, you are expected to keep regular versions of your own code in relation to the ongoing project. You will create a repository in GitHub that follows the guidelines seen in class. It should contain a Readme file and proper messages for each commit. Commits should happen on a regular basis and be properly commented on. Your repository should also be as secure in a reasonable way, depending on your project.

Time to complete: This task is to be worked out on a weekly basis, and final deliverables (including code and documents) should be submitted by the end of the course (a total of 9 weeks).

Documentation requirements:

- Please follow through with the guidelines given during class.
- There is no minimum or maximum length of the documentation, but it should follow established guidelines (see references below).

Resource:

- How to write Readmes for repositories: https://www.freecodecamp.org/news/how-to-write-a-good-readme-file/
- How to effective commit messages: https://cbea.ms/git-commit/
- How to secure repositories: https://learn.microsoft.com/en-us/training/modules/maintain-secure-repository-github/

Passing criteria: 80% (16/20 pts)

Rubric:

SLO 3 Rubric					
Criteria	**Ratings**				**Pts**
Repository follows guidelines	**5 to > 4 pts** **Excellent** Does repository have a readme? Is it following naming conventions, does it have a changelog and a requirements file, etc.	**4 to > 2 pts** **Proficient** Most guidelines are followed but not all of them.	**2 to > 1 pts** **Limited** Most guidelines are not followed	**1 to > 0 pts** **Poor** No guidelines are followed	**5**
Repository secure	**5 to > 4 pts** **Excellent** Does the repository have the right permissions for the right users set.	**4 to > 2 pts** **Proficient** Some users have excessive permissions.	**2 to > 1 pts** **Limited** Most users have excessive permissions	**1 to > 0 pts** **Poor** Repository is not secured.	**5**
Repository properly described	**5 to > 4 pts** **Excellent** Readme.md present, properly formatted and complete	**4 to > 2 pts** **Proficient** Readme.md present, complete but not formatted	**2 to > 1 pts** **Limited** Readme.md present but incomplete	**1 to > 0 pts** **Poor** No Readme.md	**5**
Commits properly described	**5 to > 4 pts** **Excellent** All commits have meaningful messages	**4 to > 2 pts** **Proficient** Most commits have meaningful messages.	**2 to > 1 pts** **Limited** Most commits have no message or are not informative	**1 to > 0 pts** **Poor** No messages in commits.	**5**
				Total Points:	**20**

Assignment 2: Join an Open Source Project and Contribute to it.

Student Learning Outcome 4: Evaluate the advantages and challenges of open-source software development.

Objective: The purpose of this assignment is for you to experience joining an Open Source Project and attempt to contribute to it as part of gaining real world experience in coding and project management.

Task: In this task, you are expected to research several Open Source Projects and select one to join. Success in joining and contributing will depend on the project, but the important part is to go through the process. You are expected to research the selected project, learn about its policies and goals, and analyze both its opportunities and limitations. You are expected to present a report and to select your project for the rest of the course based on this research. You are also expected to review the licensing options and select a proper one for your project.

Time to complete: This task is to be worked out on a weekly basis, and final deliverables (including code and documents) should be submitted by the end of the course (a total of 9 weeks).

Documentation requirements:

- Please follow through with the guidelines given during class.
- There is no minimum or maximum length of the documentation, but it should follow established guidelines (see references below).

Resource:

- List of friendly Open Source Projects: https://github.com/MunGell/awesome-for-beginners
- Information about open source licenses: https://opensource.org/licenses/#:~:text=Open%20source%20licenses%20are%20licenses,Source%20Initiative's%20license%20review%20process.

Passing criteria: 80% (16/20 pts)

Rubric:

Evaluate Open Source Projects
Rubric

Criteria	Ratings				Pts
Relevant Open Source Projects (OSS) properly identified	**5 to > 4 pts** **Excellent** Document describing each of the OSS projects evaluated with comprehensive information	**4 to > 2 pts** **Proficient** Document listing OSS projects but with not much explanation	**2 to > 1 pts** **Limited** Only one OSS project evaluated	**1 to > 0 pts** **Poor** No evidence of an OSS researched	**5**
Proper Licensing Researched and Selected	**5 to > 4 pts** **Excellent** Document describing the different licenses evaluated and justifying the one selected.	**4 to > 2 pts** **Proficient** Appropriate license selected but not a lot of evidence of further research.	**2 to > 1 pts** **Limited** A license selected but no evidence of understanding the implications	**1 to > 0 pts** **Poor** No license selected	**5**
OSS policies reviewed	**5 to > 4 pts** **Excellent** Complete presentation of policies and licenses of the selected OSS	**4 to > 2 pts** **Proficient** Some description of the policies of the OSS	**2 to > 1 pts** **Limited** Overview of the OSS	**1 to > 0 pts** **Poor** No evidence of any analysis on the policies of the OSS	**5**
Opportunities and Challenges identified	**5 to > 4 pts** **Excellent** Document describing the strengths and limitations of the selected OSS	**4 to > 2 pts** **Proficient** Some description of the opportunities and challenges of the selected OSS.	**2 to > 1 pts** **Limited** Very brief description of the opportunities	**1 to > 0 pts** **Poor** No evidence of any research done on the OSS.	**5**
				Total Points:	**20**

Assignment 5: Final Reflection

Student Learning Outcome 5: Reflect on the competencies and skills gained and discuss how earning the Micro-Credential will impact career trajectory.

Objective: The purpose of this assignment is to reflect on the experiences and knowledge acquired during this course and how it will impact your career and internship opportunities.

Task: In this task, you are expected to write a final reflection describing your experiences, how do you think the topics learned during this course will positively impact your career and to analyze further opportunities.

Time to complete: This task is to be completed in the final week of the course. You will have 1 week to deliver a document with the reflection.

Documentation requirements:

- Please follow through with the guidelines given during class.
- The reflection should be at least 2000 words in length.

Passing criteria: 80% (12/15 pts)

Rubric:

Reflection Rubric					
Criteria	**Ratings**				**Pts**
Depth of Reflection	**5 to > 4 pts** **Excellent** Response demonstrates an in-depth reflection on the skills required to be successful in a Software Developer Internship. Detailed examples are provided about what experiences will be more useful in the upcoming future.	**4 to >2 pts** **Proficient** Response demonstrates a general reflection on the experience of a possible Software Developer Internship.	**2 to >1 pts** **Limited** Response demonstrates a minimal reflection on and non-relevant examples are provided on how this course will help in the upcoming internships.	**1 to >0 pts** **Poor** Response demonstrates a lack of reflection or understanding about the key features and skills necessary in a Software Developer Internship.	**5**
Structure	**5 to > 4 pts** **Excellent** Writing is clear, concise, and well organized with excellent sentence/paragraph construction. Thoughts are expressed in a coherent and logical manner. There are no more than three spelling, grammar, or syntax errors per page of writing.	**4 to >2 pts** **Proficient** Writing is mostly clear, concise, and well organized with good sentence/paragraph construction. Thoughts are expressed in a coherent and logical manner. There are no more than five spelling, grammar, or syntax errors per page of writing.	**2 to >1 pts** **Limited** Writing is unclear and/or disorganized. Thoughts are not expressed in a logical manner. There are more than five spelling, grammar, or syntax errors per page of writing.	**1 to >0 pts** **Poor** Writing is unclear and disorganized. Thoughts ramble and make little sense. There are numerous spelling, grammar, or syntax errors throughout the response.	**5**
Evidence and Practice	**5 to >4 pts** **Excellent** Response shows strong evidence of having completed and understood the activities presented during the course	**4 to >2 pts** **Proficient** Response shows evidence of having participated in the activities.	**2 to >1 pts** **Limited** Response shows little evidence of having understood and taking advantage of the opportunities created during the course.	**1 to >0 pts** **Poor** Response shows no evidence of having participated or learned any of the tools provided during the course.	**5**
					Total Points: 15