THE LINUX FOUNDATION | Education

20250101

General Criteria and Application

Institution & Program Name

Name of the Academic Institution and Department/program completing this application

Response

Applicant Name

Name of person completing this application that will be the primary point of contact for the review process

Response

Students

Approximately how many students graduate from each related degree program every year?
(We may use this number in our own statistics about the reach of this certification program.)

Response

Curriculum

Provide the URL to the web site where information of the academic program can be found

Response

Faculty

Who is the institutional point of contact / sponsor for this effort? In general, we prefer a permanent faculty member or staff member to fulfill this role. A student is not eligible.

Response

Provide details about the open source expertise of the faculty members, staff, or teaching/research assistants.

Response

Facilities

Please provide information about the use of Linux Foundation training materials by participants at your university? (Note, these is mostly geared toward professional development) If so, how do you use it? Are there materials which you feel would be useful which are not provided?

Response

Signatories

University Leadership

Please have your department head, dean, or provost provide a letter(s) of support for this application (see Appendix for example letter).

Accreditation Relationship Representative

As the person who will be representing the accreditation relationship, please sign below attesting your support and accuracy of the application.

Printed Name: Name Here

Date: Date Here

By signing below and submitting this form to The Linux Foundation:

- The Institution agrees to the Terms and Conditions of the Accelerating Accreditation Program (the "Program Terms"), available at https://www.linuxfoundation.org/academic-computing-accreditation/terms.
- The Institution confirms that the academic program described above meet all of the requirements described herein.
- The Institution agrees that it will either (a) maintain conformance of its approved academic programs with future changes to the requirements of this Program, or (b) cease use of the Conformance Marks at the end of the applicable conformance time period described in the Program Terms.
- The Institution confirms that it will promptly submit an updated Application to The Linux Foundation prior to using the Conformance Marks with academic programs not listed here.
- I confirm that I am authorized to make the above statements and to submit this form on behalf of the Institution.

Signature:

X ______________________

THE LINUX FOUNDATION | Education

CNCF Program Specific Criteria and Application

CLOUD NATIVE COMPUTING FOUNDATION

For each of the following areas of evaluation, there should be evidence that this topic is covered in existing courses, a plan to improve, or both:

- For evidence in existing courses, this could include 1) an excerpt from a syllabus that discusses the course, 2) a series of slides (around 10) which have been used to cover the related topic, 3) an assignment or lab which gives hands-on experience with the topic, and/or 4) one or more relevant question(s) from the tests, quizzes or other assessments. It is understood that most topics will not be represented in all three ways. Please provide succinct, compelling evidence you have at hand for whichever topics you cover. Attach additional documents, as needed.
- For a plan to improve, briefly (in about 1 paragraph) describe a continuous improvement plan to add materials on this topic in the future. It is expected that some number of areas of evaluation will necessitate such a plan

At least 3 of items 1, 3, 5, and 7 should have some hands-on experience for students as is evidenced in a lab or assignment.

It is suggested (but not required) that all of the above should have an identified learning objective which is tested.

1. Virtualization & Containerization

Understand the concepts behind virtualization as an abstraction layer. Learn how to containerize an application and distinguish between use cases for containers, bare metal, and virtual machines.

Response

2. Cloud Native Architectures and Data Management

Understand cloud native architectural principles such as microservices and event-driven architectures. Learn data management strategies for both relational and non-relational systems, and the handling of persistent versus ephemeral storage.

Response

3. Security and Observability

Understand core security practices necessary for cloud native environments, including trust provisioning and secret management. Learn fundamental principles of monitoring and observability to maintain operational health and performance of applications.

Response

4. Computer Science Basics

Ensure coverage of fundamental topics such as queueing and aspect-oriented programming.

Response

5. Web Development Basics

Learn about DNS, ports, and network layers, with a focus on layers 4 and above.

Response

6. Orchestration

Understand why automation is crucial for applications, covering topics like configuration as code, declarative infrastructure, and CI/CD pipelines.

Response

7. Persistence

Understand the basics of data persistence and the differences between file and block storage, as well as SQL and NoSQL databases.

Response

8. Monitoring & Debugging

Gain basic knowledge of monitoring and observability tools for the cloud (e.g. Grafana) and understand the complexities of debugging distributed/cloud systems.

Response

OpenSSF Program Specific Criteria and Application

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

For each of the following areas of evaluation, there should be evidence that this topic is covered in existing courses, a plan to improve, or both:

- For evidence in existing courses, this could include 1) an excerpt from a syllabus that discusses the course, 2) a series of slides (around 10) which have been used to cover the related topic, 3) an assignment or lab which gives hands-on experience with the topic, and/or 4) one or more relevant question(s) from the tests, quizzes or other assessments. It is understood that most topics will not be represented in all three ways. Please provide succinct, compelling evidence you have at hand for whichever topics you cover. Attach additional documents, as needed.
- For a plan to improve, briefly (in about 1 paragraph) describe a continuous improvement plan to add materials on this topic in the future. It is expected that some number of areas of evaluation will necessitate such a plan

At least 3 of items 4, 6, 7, and 10 should have some hands-on experience for students as is evidenced in a lab or assignment.

It is suggested (but not required) that all of the above should have an identified learning objective which is tested.

1. Security Basics and Requirements

Understand the definition of security, privacy, risk management, common security requirements, secure by design, and secure by default.

Response

2. Broader Implications of Security Issues

Understand the ethical, legal, and regulatory concerns that may impact software security.

Response

3. Secure Design

Learn & follow design principles, such as least privilege, non-bypassability, least common mechanism (risk of sharing), simplicity of mechanism, fail-safe defaults, and least astonishment.

Response

4. Security Design Assessment Fundamentals

Understand how to assess the security benefits and risks of a security design, including how to perform threat modeling.

Response

5. Evaluating Third-Party Software Supply Chain Security

Understand how to assess the security of a third-party software project, such as a potential dependency.

Response

6. Secure Implementation

Learn the most common kinds of vulnerabilities & how to prevent them, e.g., buffer overflows, heap memory attacks, SQL injection, Cross-Site Scripting (XSS), and the use of memory-safe programming languages.

Response

7. Secure Verification

Understand the diverse types of tools and their use. For example, Quality scanners (linters); Static Application Security Testing (SAST); Secret scanners; Software Component Analysis (SCA)/Dependency Analysis tools (e.g., dependabot on GitHub); Fuzzers; web application scanners.

Response

8. Enabling Updating

Understand why software updates are essential, how to securely update software, and the value of this being done by default.

Response

9. Supporting Vulnerability Reports
Understand how to report a vulnerability, accept reports, and process vulnerability reports.

Response

10. Securing Project Infrastructure
Understand how to secure the source, build, test, and distribution infrastructure, including configuration of internal or external services, to set up a software project to follow security best practices. This includes topics like protecting accounts with MFA, using and enforcing second party code review, software supply chain attestations, securely using VCS tools like git, establishing security-focused forge policy, Supply chain Levels for Software Artifacts (SLSA) levels, configuring CI/CD systems to securely run tools and tests automatically (without publicly exposing secrets).

Response

Appendix

Sample Letter of Support

[OpenSSF|CNCF] Submissions

[University Letterhead]
[Date]

To Whom It May Concern,

I am writing to express my full support for the application materials submitted by [University Name] for the [Program Name] certification program initiated by the [Open Source Security Foundation | Cloud Native Computing Foundation], a project of The Linux Foundation. Our institution is committed to providing our students with the highest standards of education, and we believe that this accreditation will enhance their learning experience for our students. We appreciate the opportunity to participate in this initiative and support the application submitted by [University Name] for [Program Name].

Sincerely,
[Name]
[Title]
[University Name]