Course Outline Report

MC New Course: CS 001 - Introduction to Open Source Software and Computer Science

COLLEGE:
Merritt College

ORIGINATOR: Brown, Courtney

DIVISION/DEPARTMENT:
Merritt - Division II/M - Technology

STATE CONTROL NUMBER:

DATES:
BOARD OF TRUSTEES APPROVAL DATE:
STATE APPROVAL DATE:
CURRICULUM COMMITTEE APPROVAL DATE:
REQUISITE VALIDATION:

CURRENT EFFECTIVE DATE:

1. REQUESTED CREDIT CLASSIFICATION:

D - Credit - Degree Applicable

N - Not Basic Skills

1 - Program Applicable

2. DEPT/COURSE NO:

CS 001

3. COURSE TITLE:

Introduction to Open Source Software and Computer Science

4. COURSE:

Course
MC New Course

TOP NO. 0707.00 - Computer Software Development*

5. UNITS:

Variable No

Units (Min) 3.000

Min Total

Hours
Lecture Hours (Min) 2.000

35

Lab/Studio/Activity Hours (Min) 3.000

52.5

6. SELECTED TOPIC:

NO. OF TIMES OFFERED AS SELECTED TOPIC:

AVERAGE ENROLLMENT:

7. JUSTIFICATION FOR COURSE:

Free and Open Source Software (F/OSS) provides low cost access to the digital tools used in the creation, testing, and global deployment of software projects. It provides a way to explore interests in Computer Science, master the tools used in degree applicable classes prior to enrolling. General knowledge of how to work with F/OSS practices is often the start of a career in Software Development. Every company uses F/OSS projects, and expertise in select F/OSS projects can open the door to career opportunities worldwide. Topics include introduction to concepts and methodologies of Computer Science, a survey of projects in the Open Source Software ecosystem, and the configuration of a computer workstation that enables participation in F/OSS software development. Transferable to the CSU/UC systems and meets AA/AS area 4c requirements aligned with C-ID 112.

8. COURSE/CATALOG DESCRIPTION

Fundamental concepts in software creation: The software development life cycle, use of Free/Open Source Software (F/OSS) tools to create your own software development workstation. Introduction to Computer Science programming methodologies, software implementation techniques, coding conventions, procedural programming, object-oriented programming, testing, and algorithms. Applied use of tools to common software projects, practices useful for Computer Science major courses, or participation in open source software projects.

9. OTHER CATALOG INFORMATION

a. Modular: No

If yes, how many modules:

b. Open entry/open exit: No

c. Grading Policy: Both Letter Grade or Pass/No Pass

d. Eligible for credit by Exam: No

e. Repeatable according to state guidelines: No

f. Required for degree/certificate (specify):

g. Meets GE/Transfer requirements (specify): See GE tab

h. C-ID Number: ITIS 120

Expiration Date:

i. Are there prerequisites/corequisites/recommended preparation for this course? No

10. LIST STUDENT PERFORMANCE OBJECTIVES (EXIT SKILLS):

If an Objective cannot be deleted, make sure a Content-Review found in the Content Validation Page is not using that objective.

Objectives

1. **Describe the software development life-cycle.**
2. **Describe the principles of structured programming.**
3. **Design, implement, and test structured programs.**
4. **Explain what an algorithm is and its importance in computer programming;**

11. COURSE CONTENT:

LECTURE CONTENT:

1. Tools of open source software development:Linux, editors and Integrated Development Environment (IDE), the compiler, linker, library tool chain, autotools/make (15%)
2. Basic syntax and semantics of a higher-level language (10%)
3. Variables, types, expressions, and assignment (10%)
4. Input, Output (I/O) and Files (10%)
5. Conditional and iterative control structures (10%)
6. Functions and parameter passing (10%)
7. Pseudocode, Design, and Implementation of algorithms (10%)
8. Debugging (5%)
9. Procedural languages (10%)
10. Object-oriented languages (10%)

LAB CONTENT:

Configuring a personal Open source Software workstation (20%)

Several programs illustrating one or more topics in control structures. (15%)

Several programs illustrating one or more topics in Objects and Object Oriented Programming. (15%)

Autotools (20%)

Management and installation of library dependencies (15%)

Best practices in online discussions and threaded forums.(105%)

12. METHODS OF INSTRUCTION (List methods used to present course content):

- Activity
- Lecture
- Lab
- Observation and Demonstration
- Discussion
- Critique
- Projects
- Individualized Instruction
- Threaded Discussions

Other Methods:
Readings and exercises in textbook Programming Solution demonstrations Lab and programming assignments Team programming projects.

13. ASSIGNMENTS

Out-of-class Assignments (List all assignments, including library assignments. Requires two (2) hours of independent work outside of class for each unit/weekly lecture hour. Outside assignments are not required for lab-only courses, although they can be given.)

Override Outside Class Hours: No

Outside-of-Class Hours (Min) 4.000

Outside-of-Class Hours (Max) 0.000

Override Outside-of-Class Hours (Min) 0.000

Override Outside-of-Class Hours (Max) 0.000

Out of class Assignment
1. Assigned text readings.
2. Exercises from the textbook.
3. Programming assignments involving functional programming, flow charting, and program debugging.
4. Team programming projects.

14. STUDENT ASSESSMENT: (Grades are based on):

- ESSAY (Includes "blue book" exams and any written assignment of sufficient length and complexity to require students to select and organize ideas, to explain and support the ideas, and to demonstrate critical thinking skills.)
- COMPUTATION SKILLS
- NON-COMPUTATIONAL PROBLEM SOLVING (Critical thinking should be demonstrated by solving unfamiliar problems via various strategies.)
- SKILL DEMONSTRATION
- MULTIPLE CHOICE

OTHER (Describe):

15. TEXTS, READINGS, AND MATERIALS

A. Textbooks:
YesNo37
Walter Savitch, Kenrick Mock. *Problem Solving with C++*. 10th Edition edition Pearson, 2018.
Deitel, H.M. & Deitel, P.J.. *C++ : How to Program*. 10th Edition edition Pearson, 2017.

*Date is required: Transfer institutions require current publication date(s) within 5 years of outline addition/update.

B. Additional Resources:
Library/LRC Materials and Services:

The instructor, in consultation with a librarian, has reviewed the materials and services of the College Library/LRC in the subject areas related to the proposed new/updated course
Print Materials were reviewed? Yes

Non-Print Materials were reviewed? No

Online Materials were reviewed? Yes

Services were reviewed? Yes

Specific materials and/or services needed have been identified and discussed. Librarian comments:
The library provides sufficient information resources in both print and electronic format to support this course. A librarian is available in person at the reference desk or online via chat to assist students whenever the library is open.

C. Readings listed in A and B above are: (See definition of college level):
YesNo39
Primarily college level

16. DESIGNATE OCCUPATIONAL CODE:

C - Clearly Occupational

17. LEVEL BELOW TRANSFER:

Y - Not applicable

18. CALIFORNIA CLASSIFICATION CODE:

Y - Credit Course

19. NON CREDIT COURSE CATEGORY:

Y - Not Applicable, Credit course

20. FUNDING AGENCY CATEGORY:

Y - Not Applicable (funding not used to develop course)

REQUISITES AND ADVISORIES

STUDENT LEARNING OUTCOMES

1. **Select and implement a tool chain that can create software for the Linux operating system.**

   Project to install Editor and/or Integrated Development Environment (IDE) compiler(s), linker, and debugger.

2. **Select and configure appropriate open source libraries to complete a programming project.**

   Determine dependencies and install required libraries using package manager(s) and/or build libraries from source.

3. **Design and implement programs introducing computer sicence methjodologies and data structures.**

   Completion of a set of programming assignments.