**COMPUTER SCIENCE (CSC) PROGRAM PROJECT PRESENTATION EVALUATION FORM**

The attached evaluation tool (rubric) is designed to assist in the evaluation of students’ ability to successfully prepare and defend their projects. The rubric includes four evaluation criteria. Evaluation of a project is an integral part of student learning outcomes assessment conducted by the CSC program.

This evaluation tool will:

* Provide students, prior to their defense, with a clear understanding of the elements of their project outcome and its defense deemed most important to the program.
* Provide multiple perspectives on students’ ability to successfully prepare and defend their project and engage in cogent discourse about their chosen field of study
* Serve as a potential source of program-level data on the accomplishment of the program’s learning outcome objectives, for submission as part of an assessment report

# Suggested Instructions:

Faculty members and students should review and become familiar with the criteria in the evaluation tool prior to the project defense. The rubric should be scored at the conclusion of the defense, or shortly thereafter, by every member present. This cover page (page 1) can then be completed (providing a *summary of the scored ratings* below for each of the criteria in the rubric), returned to the appropriate program office, and maintained in a confidential departmental file following the defense (one cover page per evaluator) for use as a valuable tool in student learning outcomes assessment. The remaining rubric (page 2) can be shared with the student or destroyed.

# Student name (Please Circle One)

|  |  |  |  |
| --- | --- | --- | --- |
| Bixheku Vasil | Herrera Richard | Kuchekulla Kapil | Tapia Eric |
| Celis Francelia | Hussein Salam | Preston Michael | Thin Ben |
| Duran Marcos | Jungo Mark | Saidov Sarvar | Toshpulotov Sukhrobjon |
| Herrera Angela | Korrapati Ashok | Stec Jacob | Wang Qi |
| Yu Haoyong |  |  |  |

**Program:** Computer Science

**Date of Defens**e: Monday, May 03, 2021

# Evaluator role:

Advisor

Computer Science faculty member

Mathematics faculty member \* Student

Other (explain)

# Defense Score Summary by Criterion (From page 2): Assessment Criteria: 1: 3

4: 2

2: 3 Total: 12

3: 4

**COMMENTS:**

1. **Could have improved in Finite state machine diagram.**
2. **Needs improvement in data flow diagram**
3. **Could have included future enhancements**

Page 1 Apr 2018

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Assessment Criteria** | **4= Exemplary** | **3= Competent** | **2= Developing** | **1=Unacceptable** | **N/A** | **Score** |
| Faculty assessment of Final Presentation and Demo | | | | | | |
| 1. The student makes effective | The student speaks | The student speaks using | The student speaks | The student doesn’t speak |  | 3 |
| technical oral presentations to | loudly, and clearly | good grammar. The | using good grammar. | well. The materials are |
| faculty and staff. | using good grammar | presentation is organized | The presentation can | distracting and/or largely |
|  | and professional tone. | and the materials support | be improved through | non-supportive of the |
|  | The presentation is well | the presentation. | reorganization and/or | presentation. |
|  | organized and materials | Responds to questions. | improved materials. |  |
|  | effectively support the |  |  |  |
|  | presentation. |  |  |  |
|  | Responds to questions |  |  |  |
|  | well. |  |  |  |
| 2. The student applies modern | The student performs | The student develops | The student develops | The student develops code |  | 3 |
| design techniques to ensure | competently and in | code that follows the | code that follows the | without following the |
| code meets design | addition notices flaws | design spec, is designed | design spec, is designed | design spec and/or without |
| specifications. | in or improvements that | based on structured and | based on structured and | using structured and |
|  | can be made to the | OO programming | OO programming | OO programming |
|  | design spec and | techniques, and utilizes | techniques, but often | techniques. The code must |
|  | consistently delivers | design patterns where | must be revised before it | usually be rewritten by |
|  | code that is of | appropriate. | is acceptable. | others. |
|  | exceptional quality. |  |  |  |
| 3. Imagination and originality | Problem/purpose of | Problem/purpose of | Problem/purpose of | Problem/purpose of study |  | 4 |
| of thought | study very creative or | study original or | study moderately | lacked creativity or not |
|  | original with new and | creative; | original or creative; | new; Duplication of |
|  | innovative ideas; | Design/approach | Design/ approach | previous work. |
|  | Explored original topic | appropriate or | moderately appropriate |  |
|  | and discovered new | innovative. | or innovative. |  |
|  | outcomes. |  |  |  |
| 4. The student performs thorough testing of developed  software system. | The student plans and executes  thorough list of | The student plans and executes thorough list of  test cases for black‐box | The student performs  black‐box unit testing of own code, using | The student per‐forms minimal unit testing of own  code, concentrating |  | 2 |
|  | test cases for  black‐box unit and | testing of his/her own  code as well as the | reasonable sample of  average and extreme | exclusively on the simplest,  most obvious test cases. |
|  | system testing, as well  as white‐box testing of | team’s integrated code,  with expected results | test cases. |  |
|  | modules. | specified. |  |  |

Page 2