**Subject: Advance Programming**

**Syllabus**

**PROFESSOR: Luis Gerardo Cámara Salinas**

**Engineering Program: EMBEDDED SYSTEMS ENGINEERING**

|  |  |  |  |
| --- | --- | --- | --- |
| LEARNING PURPOSE OF THE SUBJECT | The student will develop software, implementing models, development  methodologies, test methodologies, for its application to embedded systems. | | |
| QUARTER | Third | | |
| TOTAL HOURS | 90 | **WEEKLY HOURS** | 6 |

|  |  |  |
| --- | --- | --- |
| LEARNING UNITS | TOTAL HOURS | PICE SUBMISSION DEADLINE |
| Software Development Models | 11 | **17/05/2021** |
| Software Development | 30 | **25/07/21** |
| Software Quality Methods | 25 | **06/08/2021** |
| Software Implementation | 24 |  |
| TOTAL | **90** |  |

**Grading Policy**

**(From Doc.: Academic Guidelines)**

* Evaluation scale is twofold:
  1. Competent (approved):

1. Competent
2. Independent
3. Advanced Basic
4. Basic

II. Not Competent (not approved).

According to the next criteria:

|  |  |
| --- | --- |
| **Achieved Grade** | **Registered Grade** |
| 0.0 to 6.99 | 6 Not Competent (NC) |
| 7.0 to 7.49 | 7 Basic (BU) |
| 7.50 to 8.49 | 8 Advanced Basic (BA) |
| 8.50 to 9.49 | 9 Independent (I) |
| 9.50 to 10.00 | 10 Competent (C) |

* The student must approve every single learning unit in order to approve the subject.
* Types of Assessment

1. Ordinary;
2. Extemporaneous;
3. Extraordinary;
4. Special.

**Teacher’s contact information**

E-mail: luis.camara@upy.edu.mx

Red Virtual UPY: lcamara

**Schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| Class Schedule | Monday  9:50 – 11:30 hrs | Tuesday  12:30 – 14:10 hrs | Thursday  12:30 – 14:10 hrs |

**LEARNING UNITS**

|  |  |
| --- | --- |
| LEARNING UNIT | I. Software Development Models |
| EXPECTED OUTCOME | The student will adapt development models to outline software projects. |
| HOURS | 11 |

|  |  |  |  |
| --- | --- | --- | --- |
| TOPICS | KNOWLEDGE | KNOW-HOW | SESSION DATES |
| Software  development  life cycle | Define the concept of life cycle. Describe the elements of the life cycle: requirements  analysis, design, implementation, verification and maintenance | Outline software development  projects. | Week 1 |
| Software  development  models | Describe the characteristics of software development models. Explain the  characteristics of software development models: Prototype model Spiral model  Development model by stages Incremental or iterative model Waterfall model | Adapt project development  schemes to software development  models. | Week 2 |

|  |  |  |
| --- | --- | --- |
| ASSESSMENT PROCESS | | |
| PERFORMANCE EVIDENCE | **%** | **ASSESSMENT INSTRUMENTS** |
| Portfolio | 30 | Rubric Checklist |
| Project | 50 | Rubric Checklist |
| Quiz | 20 | Rubric Checklist |
| TOTAL | 100% |  |

|  |  |
| --- | --- |
| LEARNING UNIT | II. Software Development |
| EXPECTED OUTCOME | The student will plan the development of the software, for its implementation. |
| HOURS | 30 |

|  |  |  |  |
| --- | --- | --- | --- |
| TOPICS | KNOWLEDGE | KNOW-HOW | SESSION DATES |
| Software  development  philosophies | Explain software development philosophies: Lean and  Agile. | Develop software according to the agile  philosophy. Develop software according to the lean  philosophy. . | Week 3 |
| Software  Development  Methodologies | Explain software development methodologies: Scrum,  Kanban, XP and TPS. | Develop software according to software development  methodologies. | Week 4 |
| Software  development  cycle tools | Explain the tools of the software development cycle:  Sprints, Boards, Cohort Analysis and Pair  Programming. | Implement the tools of the software development cycle. | Week 5, Week 6 |

|  |  |  |
| --- | --- | --- |
| ASSESSMENT PROCESS | | |
| PERFORMANCE EVIDENCE | **%** | **ASSESSMENT INSTRUMENTS** |
| Portfolio | 30 | Rubric Checklist |
| Project | 50 | Rubric Checklist |
| Quiz | 20 | Rubric Checklist |
| TOTAL | 100% |  |

|  |  |
| --- | --- |
| LEARNING UNIT | III. Software Quality Methods |
| EXPECTED OUTCOME | The student will develop a program data protection, to mitigate and minimize ries g os in databases. |
| HOURS | 25 |

|  |  |  |  |
| --- | --- | --- | --- |
| TOPICS | KNOWLEDGE | KNOW-HOW | SESSION DATES |
| Validation  Techniques | Identify software validation techniques. Explain software validation  techniques: - Equivalent partitions - Edge | Validate developed software | Week 7 |
| Software Quality  Testing | Explain the operation and syntax of input and output instructions of digital signals in the programming of rapid prototyping cards. | Validate the quality of the software  developed . | Week 8 |
| Quality Metrics | Explain the quality metrics: - Test plan - Error analysis document -  Requirements matrix - Test cases | Validate software tests | Week 9 |

|  |  |  |
| --- | --- | --- |
| ASSESSMENT PROCESS | | |
| PERFORMANCE EVIDENCE | **%** | **ASSESSMENT INSTRUMENTS** |
| Portfolio | 30 | Rubric Checklist |
| Project | 50 | Rubric Checklist |
| Quiz | 20 | Rubric Checklist |
| TOTAL | 100% |  |

|  |  |
| --- | --- |
| LEARNING UNIT | IV. Software Implementation |
| EXPECTED OUTCOME | The student will implement libraries and subroutines to patent the developed software |
| HOURS | 24 |

|  |  |  |  |
| --- | --- | --- | --- |
| TOPICS | KNOWLEDGE | KNOW-HOW | SESSION DATES |
| Specialized  libraries | Explain the characteristics of specialized libraries and subroutines. | Implement libraries and subroutines. | Week 10 |
| Software  licensing and  intellectual  property | Describe the elements of the software license. Describe the software license  classification. | License software development patent processes. | Week 11, Week 12 |

**CODE OF CONDUCT IN THE CLASSROOM**

Mutual respect

Honesty

Integrity

Listen closely to classmates at exhibitions

Empathy

**BIBLIOGRAPHIC REFERENCES**

|  |  |  |  |
| --- | --- | --- | --- |
| AUTHOR | YEAR | DOCUMENT TITLE | PLACE OF PUBLICATION |
| Zed A. Shaw | 2005 | Learn C the Hard Way | US |
| Delores etter | 2008 | Engineering Problem Solving with C | US |
| Brian W. Kernighan  Dennis M. Ritchie | 2003 | *The C Programming Language* | US |
| Steve Oualline | 1997 | *Practical C Programming* | US |
| Silberschatz, A. | 2014 | Database Fundamentals | US |