## A COURSE MODULE DESCRIPTOR FORM

|  |  |  |  |
| --- | --- | --- | --- |
| Module Information | | | |
| **Course Module Title** | Mobile Application | | |
| **ناوى کۆرس مۆدیول** | ژمێریاری مۆبایلی جوڵاو | | |
| **عنوان الوحدة** | حوسبة الهاتف النقال | | |
| **Course Module Type** | Core | **Module Code** | **IT402** |
| **ECTSs** | 4 | | |
| **Department** | Information Technology Department | | |
| **Department Code** | IT | | |
| **Module Website (CMW)** | https://noble.edu.krd/lms/login.php | | |
| **Module Leader (ML)** | Vian Waheed Khalid | | |
| **NTI - E - mail** | vian.wahid@noble.edu.krd | | |
| **ML Acad. Title** | Assistant Lecturer | | |
| **ML ORCID** | 0000-0003-4732-0477 | | |
| **ML Google Scholar Acc** | https://scholar.google.com/citations?hl=en&user=VPGeNpMAAAAJ | | |

|  |  |  |
| --- | --- | --- |
| Relation with Other Modules | | |
| **Pre-requisites** | **Data Structures** |
| Module Aims, Learning Outcomes and Indicative Contents | | |
| Module Introductory Description | Mobile Computing is a technical field that covers the design, development, and evaluation of mobile applications using appropriate solutions that meet user requirements. This includes learning the technology used to perform a wide variety of tasks on portable devices e. Portable devices include Smart Phones, Tablets, Laptops, wearable devices, vehicles, etc.  Upon completion of this specialization, students should be able to develop mobile apps applying algorithmic and programming concepts that are cross-platform beyond Android |
| Module Aims | 1. Introduce the basic principles of Operating Systems and computer architectures coupled together with aspects like process management, memory management, file systems, I/O subsystems, etc. 2. Practical application of OS principles by examining case studies in mobile platforms focusing on the use of Android API. 3. Experience of using a development environment and emulator for mobile devices. 4. Provide insights in more advanced Android programming principles and experience in developing and maintaining efficient Android system applications that fully leverage the OS principles taught. 5. Develop the students' understanding of OS issues (i.e., process and thread management, resource management and communication, etc.) relating to embedded systems such as mobile devices. |
| Module  Learning Outcome | On successful completion of this course, each student is able to:   1. Understand the fundamental of Operating System principles, abstractions, mechanisms and their implementations. 2. Design, develop and test a working application on a mobile device focusing on the use and primary functionalities of the Android OS. 3. Utilize the advanced features of Android OS in the context of how memory is managed, how tasks are scheduled, how interrupts are handled, file systems, I/O subsystems, etc. and put these concepts into practice by building (optimized) Android system applications. 4. Appreciate the complexities of OS issues (i.e., process and thread management, resource management and communication, etc.) in deploying commercial mobile applications and leading-edge developments in the mobile application marketplace. 5. A comprehension and appreciation of the design and development of context-aware solutions for mobile devices. |
| Learning and Teaching Strategies | | |
| **Strategies** | Small groups, video learning, working on projects, student center (presenting seminars by students), scientific trips to telecommunication companies, letting students become an assistant at lab.  Class activities like; problem-based learning, Gap fill, Jigsaw method, Teach back teaching method.  The mentioned learning and strategies have been implemented as a strategy of learning and teaching in order to motivate the students to participate and engage to the class more effectively. |

|  |
| --- |
| **Required texts and References** |
| Introduction to Android Application Development: Android Essentials  A Hands-on Guide to Building Apps with iOS and Android. Jakob Iversen, Michael Eierman |

|  |  |
| --- | --- |
| Module Delivery | |
| **Total workload** | |
| **Contact Theoretical Hours – Per term** | 30 |
| **Contact Practical Hours – Per term** | 78 |

|  |  |  |  |
| --- | --- | --- | --- |
| Module Assessment | | | |
| **Module Activities** | **Time /Number** | **Weight (Marks)** | **Week Due** |
| Contact hours – Participation | Daily bases | 5% | Weekly |
| (Science / Lab)  (Social science / Critical thinking) | Daily bases | 5% | Weekly |
| Presentation / Seminar | 1 | 5% | 7 |
| Tutorial | 1 | 5% | 6 |
| Quiz | 2 | 5% | 4 , 11 |
| Self-study | 1 | 5% | 5 |
| Projects | 1 | 5% | 13 |
| Oral assessment | 2 | 5% | 3 , 12 |
| Midterm Exam | 1 | 20% | 8 |
| Final Exam | 1 | 40% | 15 |
| **Total** |  | 100% |  |

|  |  |
| --- | --- |
| Delivery Plan theory (Designed Syllabus) | |
|  | **Course Module Content** |
| Week 1 | Introduction to Mobile Application   * Brief history and evolution of mobile applications. |
| Week 2 | Different mobile platforms   * Android, iOS. * Operating System Concepts and Background |
| Week 3 | Mobile App Architecture   * Understanding the architecture of mobile applications. * Model-View-Controller (MVC) and other architectural patterns. |
| Week 4 | User Interface Design Principles   * UI/UX Fundamentals * Principles of effective mobile app design. |
| Week 5 | * Human-Computer Interaction (HCI) principles. * Resource Management and Communication |
| Week 6 | Mobile App Development Life Cycle   * Planning, design, development, testing, and deployment. * Agile vs. Waterfall methodologies. |
| Week 7 | Programming Languages for Mobile Development   * Introduction to languages * Cross-platform development languages (Dart, JavaScript). |
| Week 8 | **Midterm Exam** |
| Week 9 | Logic and Control Flow   * Introduction to programming logic.   Control structures (if, else, switch). |
| Week 10 | Data Handling in Mobile Apps   * Local storage and databases. * Data synchronization and security. |
| Week 11 | Mobile App Security   * Encryption. * User authentication and authorization. |
| Week 12 | Mobile App Performance Optimization   * Optimizing code for speed. |
| Week 13 | Group projects |
| Week 14 | Review |
| Week 15 | ***Final exam*** |
|  | |
| Delivery Plan practical (Designed Syllabus) | |
|  | **Course Module Content** |
| Week 1 | Introduction to MIT App Inventor interface.   * Setting up MIT App Inventor account. |
| Week 2 | Create the first interface project   * Understanding components and properties. |
| Week 3 | Designing interactive user interfaces.   * Adding buttons, textboxes, and images. |
| Week 4 | Programming Logic   * Implementing basic programming logic with blocks. * Creating decision-based applications. |
| Week 5 | Data handling   * Storing and retrieving data using local storage. * Working with variables and lists. |
| Week 6 | Multi-screen Applications   * Creating apps with multiple screens. * Passing data between screens. |
| Week 7 | Project 1 design a mobile application   * Applying what we have learned before |
| Week 8 | **Midterm Exam** |
| Week 9 | Multi-screen Applications (more details)   * Creating apps with multiple screens. * Passing data between screens. |
| Week 10 | Device Features Integration   * Utilizing device features. * Developing location-aware applications. |
| Week 11 | Advanced UI and Animation   * Implementing advanced UI features. |
| Week 12 | Advanced UI and Animation   * Enhancing user experience with creative use of available tools. |
| Week 13 | Group projects |
| Week 14 | Project 3 creating a mobile application  Applying what we have learned |
| Week 15 | ***Final exam*** |

|  |
| --- |
| Course Keywords |
| Android, iOS |