**MINISTRY OF EDUCATION, MALAYSIA**

**VOCATIONAL COLLEGE STANDARD CURRICULUM**

**COURSE INFORMATION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COURSE NAME** | **:** |  | **WEB PROGRAMMING** | |
| **CODE NAME** | **:** |  | **KPD2023** | |
| **LEVEL** | **:** |  | **3 SEMESTER 2** | |
| **CREDIT UNIT** | **:** |  | **3** | |
| **CONTACT HOUR** | **:** |  | **FACE TO FACE** | **: 6 HOURS/WEEK** |
|  |  |  | **NON FACE TO FACE** | **:** |
| **COURSE TYPE** | **:** |  | **VOCATIONAL** | |
| **PREREQUISITE**  **CORE REQUISITE** | **:**  **:** |  | **-**  **-** | |

**COURSE OUTCOMES**

At the end of the course, the students should be able to:-

* **Interpret application module development requirement**
* **Setup local environment**
* **Plan module expected behaviour**
* **Write module code**
* **Backup source code**

**COURSE DESCRIPTION**

Application module development is the process of **coding, testing, debugging & maintaining the source code**.

The person who is competent in this CU shall be able to **interpret application module development requirement, setup local environment, plan module expected behaviour, write module code and commit source code**.

The outcome of this competency is to develop an application that **functions, deployment ready, error free** and as per **Software Requirement Specification** (SRS).

**CONTENT AND LEARNING STANDARDS**

|  |  |  |
| --- | --- | --- |
| **PROGRAM** | **:** | **TEKNOLOGI PENGURUSAN PANGKALAN DATA DAN APLIKASI WEB** |
| **COURSE NAME** | **:** | **WEB PROGRAMMING** |
| **CODE NAME** | **:** | **KPD2023** |

|  |  |  |  |
| --- | --- | --- | --- |
| **CONTACT HOURS**  **(TRAINING DURATION)** | **CONTENT STANDARD**  **(WORK ACTIVITIES)** | **LEARNING STANDARD**  **(RELATED KNOWLEDGE / APPLIED SKILLS / ATTITUDE / SAFETY / ENVIRONMENTAL)** | **PERFORMANCE CRITERIA / ASSESSMENT CRITERIA** |
| **10 HOURS**  **(2 WEEKS)**  Related Knowledge  (5 Hours)  1 Week  Applied Skills  (5 Hours)  1 Week | **1.0 INTERPRET**  **APPLICATION**  **MODULE**  **DEVELOPMENT**  **REQUIREMENT** | **Related Knowledge**   1. Types of functional specification document such as:  * Functional Design Specification (FDS) * Software Design Document (SDD)  1. Function of Software Requirement Specification   (SRS)   1. Introduction to scripting language 2. Introduction to Integrated Development   Environment (IDE)   1. Introduction to Test Driven Development (TDD) 2. Project brief content such as:  * Development timeline * Modules number * Task assignation  1. Application mock up 2. Third party component such as:  * Payment gateway * Security certificate * Single Sign On (SSO)  1. Awareness of End User License Agreement (EULA) 2. Software licensing such as:  * Proprietary * Open source  1. Unified Modelling Language (UML)   **Applied Skills**   1. Obtain project brief 2. Identify module function and specification 3. Identify development timeline 4. Identify task assignation 5. Check application flow 6. Check application mock up design 7. Check third party component requirement   **Attitude:**   1. Proactive when interpreting application module development requirement 2. Resourceful when interpreting application module development requirement 3. Committed when interpreting application module development requirement 4. Analytical thinking when interpreting application module development requirement 5. Adhere to End User License Agreement (EULA)   ***Safety:***   1. Adhere to workplace ergonomics practice | **Assessment Criteria**   1. Function of Software Requirement Specification (SRS) explained 2. Types of scripting language listed 3. Function of Integrated Development Environment (IDE) explained 4. Function of Test Driven Development (TDD) explained 5. Types of project brief content listed 6. Function of application mock up explained 7. Third party component listed 8. Purpose of End User License Agreement (EULA) explained 9. Types of software licensing listed 10. Application prototype development requirement confirmed 11. Workplace ergonomic practice explained     **Performance Criteria**   * 1. Module function and specification interpreted according to Software Requirement Specification (SRS)   2. Development timeline confirmed according to job brief   3. Task assignation confirmed according to job brief   4. Application flow interpreted according to Software Requirement Specification (SRS)   5. Application mock up design interpreted according to storyboard   6. Third party component requirement (payment gateway, security certificate) interpreted according to Software Requirement Specification (SRS) |

**CONTENT AND LEARNING STANDARDS**

|  |  |  |
| --- | --- | --- |
| **PROGRAM** | **:** | **TEKNOLOGI PENGURUSAN PANGKALAN DATA DAN APLIKASI WEB** |
| **COURSE NAME** | **:** | **WEB PROGRAMMING** |
| **CODE NAME** | **:** | **KPD2023** |

|  |  |  |  |
| --- | --- | --- | --- |
| **CONTACT HOURS**  **(TRAINING DURATION)** | **CONTENT STANDARD**  **(WORK ACTIVITIES)** | **LEARNING STANDARD**  **(RELATED KNOWLEDGE / APPLIED SKILLS / ATTITUDE / SAFETY / ENVIRONMENTAL)** | **PERFORMANCE CRITERIA / ASSESSMENT CRITERIA** |
| **10 HOURS**  **(2 WEEKS)**  Related Knowledge  (5 Hours)  1 Week  Applied Skills  (5 Hours)  1 Week | **2.0 SETUP LOCAL**  **ENVIRONMENT** | **Related Knowledge**   1. Introduction to development environment 2. Introduction to local server such as:  * Local server access configuration * Installation procedure  1. Introduction to Source Code Management (SCM) such as:  * Source code repository * Branching * Revision * Access control * Distribution  1. Development stack installation procedure for:  * Windows, Apache, MySQL, PHP (WAMP)  1. Database administration such as:  * Database access * Data manipulation * Database maintenance   **Applied Skills**   1. Interpret local environment requirement 2. Interpret database structure 3. Select programming language 4. Check local server access configuration 5. Check development server access configuration 6. Check Source Code Management (SCM) access 7. Install Integrated Development Environment (IDE) 8. Install development stack (i.e. WAMP, LAMP) 9. Install local server 10. Install local database server 11. Install Source Code Management (SCM) software   **Attitude:**   1. Proactive when setup local environment 2. Resourceful when setup local environment 3. Committed when setup local environment 4. Analytical thinking when setup local environment 5. Follow company’s installation guideline   ***Safety:***   1. Adhere to workplace ergonomics practice | **Assessment Criteria**   1. Definition of development environment explained 2. Definition of local server explained 3. Definition of Source Code Management (SCM) explained 4. Development stack installation procedure explained 5. Database administration explained 6. Integrated Development Environment (IDE) installation and functionality confirmed 7. Development stack installation and functionality confirmed 8. Local server installation and functionality confirmed 9. Local database server installation and functionality confirmed 10. Source Code Management (SCM) software installation and functionality confirmed   **Performance Criteria**   * 1. Local environment requirement interpreted according to job brief   2. Database structure interpreted according to job brief   3. Language programming to be used confirmed according to job brief   4. Local server access configuration confirmed according to job brief   5. Development server access configuration confirmed according to job brief   6. Source Code Management (SCM) access confirmed according to job brief   7. Development kit ( [Integrated Development Environment](https://en.wikipedia.org/wiki/Integrated_development_environment) IDE) installed according to installation procedure   8. Development stack (WAMP, LAMP) installed according to installation procedure   9. Local server installed according to installation procedure   10. Local database server installed according to installation procedure   11. Source Code Management (SCM) software installed according to installation procedure |

**CONTENT AND LEARNING STANDARDS**

|  |  |  |
| --- | --- | --- |
| **PROGRAM** | **:** | **TEKNOLOGI PENGURUSAN PANGKALAN DATA DAN APLIKASI WEB** |
| **COURSE NAME** | **:** | **WEB PROGRAMMING** |
| **CODE NAME** | **:** | **KPD2023** |

|  |  |  |  |
| --- | --- | --- | --- |
| **CONTACT HOURS**  **(TRAINING DURATION)** | **CONTENT STANDARD**  **(WORK ACTIVITIES)** | **LEARNING STANDARD**  **(RELATED KNOWLEDGE / APPLIED SKILLS / ATTITUDE / SAFETY / ENVIRONMENTAL)** | **PERFORMANCE CRITERIA / ASSESSMENT CRITERIA** |
| **30 HOURS**  **(5 WEEKS)**  Related Knowledge  (10 Hours)  2 Weeks  Applied Skills  (15 Hours)  3 Weeks | **3.0 PLAN MODULE**  **EXPECTED**  **BEHAVIOUR** | **Related Knowledge**  3.1 Definition of module expected behaviour  3.2 Function of module test script  3.3 Test Driven Development (TDD) method such as:   * Pseudo code * Data flow diagram * Module process flow * Decision table   **Applied Skills**  3.1 List out module expected scenarios  3.2 List out module expected input and output  3.3 Write module test script  **Attitude:**   1. Proactive when planning module expected behaviour 2. Resourceful when planning module expected behaviour 3. Committed when planning module expected behaviour 4. Analytical thinking when planning module expected behaviour   **Safety:**   1. Adhere to workplace ergonomics practice | **Assessment Criteria**   1. Definition of module expected behaviour explained 2. Function of module test script explained 3. Test Driven Development (TDD) method listed 4. Module expected input and output listed 5. Module test script produced and expected functionality confirmed   **Performance Criteria**   * 1. Module expected scenarios listed out according to Software Requirement Specification (SRS)   2. Module expected input and output listed out according to Software Requirement Specification (SRS)   3. Module test script written according to Software Requirement Specification (SRS) |

**CONTENT AND LEARNING STANDARDS**

|  |  |  |
| --- | --- | --- |
| **PROGRAM** | **:** | **TEKNOLOGI PENGURUSAN PANGKALAN DATA DAN APLIKASI WEB** |
| **COURSE NAME** | **:** | **WEB PROGRAMMING** |
| **CODE NAME** | **:** | **KPD2023** |

|  |  |  |  |
| --- | --- | --- | --- |
| **CONTACT HOURS**  **(TRAINING DURATION)** | **CONTENT STANDARD**  **(WORK ACTIVITIES)** | **LEARNING STANDARD**  **(RELATED KNOWLEDGE / APPLIED SKILLS / ATTITUDE / SAFETY / ENVIRONMENTAL)** | **PERFORMANCE CRITERIA / ASSESSMENT CRITERIA** |
| **42 HOURS**  **(7 WEEKS)**  Related Knowledge  (15 Hours)  3 Weeks  Applied Skills  (20 Hours)  4 Weeks | **4. 0 WRITE MODULE**  **CODE** | **Related Knowledge**   1. Types of scripting / programming language such as:  * PHP: Hypertext PreProcessor (PHP) * Structured Query Language (SQL)  1. Programming language model such as:  * Structured programming * Procedural programming  1. Introduction to Basic Syntax  * PHP tags * Echo * Comments  1. Define Variable  * Variable * Constants * Data types * Variable Scope * Variable Variables  1. Operators  * Arithmetic Operator * Assignment Operator * Comparison Operator * Logical Operator  1. Array  * Numeric Array * Associative array  1. Control Structure  * If Else * Elseif * while Loop * Foreach Loop * Switch * Break * Continue * Include and Require  1. Function  * User-Defined Functions * Function Parameters * The Return Statement  1. Predefined Variables  * PHP Form * GET and POST  1. MySQLi  * mysqli\_fetch\_assoc() * mysqli\_fetch\_array() * mysqli\_fetch\_row()  1. Definition of user interface and user experience 2. Application mock up 3. Unit testing 4. Debugging procedure   **Applied Skills**  4.1 Check naming convention  4.2 Write instruction code to perform module function    4.3 Associate database connection with source code in  local server  4.4 Compose Structured Query Language (SQL)  statement  4.5 Develop coding according to test script  4.6 Execute unit testing  4.7 Verify module output  4.8 Debug module code  Attitude:   1. Proactive when writing module code 2. Resourceful when writing module code 3. Committed when writing module code 4. Analytical thinking when writing module code 5. Meticulous when writing module code 6. Follow company’s coding guideline   *Safety:*   1. Adhere to workplace ergonomics practice | **Assessment Criteria**   1. Types of scripting language listed 2. Programming language model listed 3. Basic Syntax introduced 4. Variable defined 5. Operators explained 6. Array described 7. Control Structure explained 8. Function explained 9. Predefined variable explained 10. MySQLi introduced 11. Definition of user interface and user experience explained 12. Function of application mock up explained 13. Definition of unit testing explained 14. Debugging procedure explained 15. Instruction code produced and functionality checked 16. Database connected with source code in local server 17. Structured Query Language (SQL) statement produced and expected functionality confirmed 18. Coding produced and expected functionality confirmed 19. Module code error fixed and expected functionality confirmed   **Performance Criteria**   * 1. Variables named according to coding/naming convention   2. Instruction code written to perform module function according to Software Requirement Specification (SRS)   3. Database connection  associated with source code in local server   4. Structured Query Language (SQL) statement written according to data flow in the Software Requirement Specification (SRS)   5. Coding developed according to test script   6. Module output verified according to Software Requirement Specification (SRS)   7. Module code debugged according to module output |

**CONTENT AND LEARNING STANDARDS**

|  |  |  |
| --- | --- | --- |
| **PROGRAM** | **:** | **TEKNOLOGI PENGURUSAN PANGKALAN DATA DAN APLIKASI WEB** |
| **COURSE NAME** | **:** | **WEB PROGRAMMING** |
| **CODE NAME** | **:** | **KPD2023** |

|  |  |  |  |
| --- | --- | --- | --- |
| **CONTACT HOURS**  **(TRAINING DURATION)** | **CONTENT STANDARD**  **(WORK ACTIVITIES)** | **LEARNING STANDARD**  **(RELATED KNOWLEDGE / APPLIED SKILLS / ATTITUDE / SAFETY / ENVIRONMENTAL)** | **PERFORMANCE CRITERIA / ASSESSMENT CRITERIA** |
| **5 HOURS**  **(1 WEEK)**  Related Knowledge  2 Hours  Applied Skills  3 Hours | **5.0 BACKUP SOURCE CODE** | **Related Knowledge**  5.1 Maintenance of source code such as:   * SCM * Version control   5.2 Source code distribution method such as:   * Pull * Push * Check out * Commit * Update   5.3 Format of work progress report  **Applied Skills**  5.1 Determine source code repository destination  5.2 Upload module source code to source code  repository  5.3 Update work progress report  5.4 Report module source code submission to superior  **Attitude:**   1. Proactive when committing source code 2. Resourceful when committing source code 3. Committed when committing source code 4. Meticulous when committing source code   **Safety:**   1. Adhere to workplace ergonomics practice | **Assessment Criteria**   1. Function of source code maintenance explained 2. Source code distribution method listed 3. Module source code transferred to source code repository   **Performance Criteria**   * 1. Source code repository destination determined according to job brief   2. Module source code uploaded to source code repository   3. Work progress report updated according to company’s Standard Operating Procedure (SOP)   4. Application source code submission reported to superior |

**Employability Skills**

|  |  |
| --- | --- |
| **Core Abilities** | **Social Skills** |
| 01.01 Identify and gather information.  01.02 Document information procedures or processes.  01.03 Utilize basic IT applications.  02.01 Interpret and follow manuals, instructions and SOP's.  02.03 Communicate clearly.  02.04 Prepare brief reports and checklist using standard forms.  02.05 Read/Interpret flowcharts and pictorial information.  03.02 Demonstrate integrity and apply practical practices.  03.03 Accept responsibility for own work and work area.  03.04 Seek and act constructively upon feedback about work performance.  03.06 Respond appropriately to people and situations.  03.07 Resolve interpersonal conflicts.  06.01 Understand systems.  06.02 Comply with and follow chain of command.  06.03 Identify and highlight problems.  06.04 Adapt competencies to new situation systems.  01.04 Analyze information.  01.05 Utilize the Internet to locate and gather information.  01.06 Utilize word processor to process information.  02.07 Utilize Local Area Network (LAN)/Intranet to exchange information.  02.08 Prepare pictorial and graphic information.  03.08 Develop and maintain a cooperation within work group.  04.01 Organize own work activities.  04.02 Set and revise own objectives and goals.  04.03 Organize and maintain own workplace.  04.04 Apply problem solving strategies.  04.05 Demonstrate initiative and flexibility.  06.05 Analyse technical systems.  06.06 Monitor and correct performance of systems.  01.07 Utilize database applications to locate and process information.  01.08 Utilize spreadsheets applications to locate and process information.  01.10 Apply a variety of mathematical techniques.  01.11 Apply thinking skills and creativity.  02.09 Prepare flowcharts.  02.10 Prepare reports and instructions.  02.11 Convey information and ideas to people.  03.15 Liaise to achieve identified outcomes.  05.01 Implement project/work plans.  05.02 Inspect and monitor work done and/or in progress. | Communication skills  Conceptual skills  Interpersonal skills  Learning skills  Leadership skills  Multitasking and prioritising  Self-discipline  Teamwork |

|  |  |
| --- | --- |
| **Tools, Equipment and Materials (TEM)** | |
| **ITEMS** | **RATIO (TEM : Trainees)** |
| 1. Computer set 2. Internet connection 3. Source Code Management (SCM) software 4. IDE software 5. Software Development Kit (SDK) 6. Database Management System (DBMS) 7. Word processing software 8. Computer with server role 9. Stationeries | 1:1  As required  1:1  1:1  1:1  1:1  1:1  1:25  As required |

|  |
| --- |
| **Reference** |
| **REFERENCES** |
| 1. Keith Cooper, Linda Torczon(2011), Engineering a Compiler (2nd Edition), Morgan Kaufmann, ISBN: 978-0120884780 2. Todd Zaki Warfel (2009), Prototyping: A Practitioner's Guide, Louis Rosenfeld Media, LLC, ISBN I-933820-21-7 3. Steve McConnell (2004), Code Complete: A Practical Handbook of Software Construction, (2nd Edition),Microsoft Press, ISBN: 978-0-7356-1967-8 4. Michael L. Scot (2009) , Programming Language Pragmatics, Third Edition, Morgan Kaufmann, ISBN-13: 978-0123745149 5. Simon Marlow (2013), Parallel and Concurrent Programming in Haskell: Techniques for Multicore and Multithreaded Programming (1st Edition), O'Reilly Media, ISBN-13: 978-1449335946 6. Scott Meyers (2005), Effective C++: 55 Specific Ways to Improve Your Programs and Designs (3rd Edition),  Addison-Wesley Professional, ISBN-13: 978-0321334879 7. Ellen Siever, Stephen Figgins, Robert Love, Arnold Robbins (2009), Linux in a Nutshell (6th Edition), O'Reilly Media, ISBN-13: 978-0596154486 |

**Disediakan oleh:**

**KEMENTERIAN PENDIDIKAN MALAYSIA**

**OCT 2018**