

DEPARTMENT OF HIGHER EDUCATION AND TRAINING

**EVALUATION REPORT FORM FOR TVET COLLEGE TEXTBOOKS**

**FORMS MUST BE COMPLETED USING THE COMPUTER.**

**ALL SECTIONS INCLUDING “NOTES” SHOULD BE COMPLETED IN FULL**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DATE |  | | TITLE NUMBER |  | |
| **1** | SCREENER | | **Combined Report** | | |
| **2** | CONTACT NUMBER | |
| **3** | LEVEL | | 2 | | |
| **4** | SUBJECT | | Basic Principles of Computer Programming and Computer Literacy | | |
| **5** | LANGUAGE | | Eng | | |
| **6** | FULL TITLE | | NCV2 Basic Principles of Computer Programming and Computer Literacy | | |
| **7** | AUTHORS | | Godwin Dzvapatsva | | |
|  | |  | | |
| **8** | PUBLISHER | | Future Managers | | |
| **9** | **ITEMS RECEIVED** | **Q**  **T** Y | **ISBN NUMBER** | | **PRICE** |
| Student Textbook | 1 | 978-0-63911-092-9 | |  |
| Lecturer Guide | 1 | 978-0-63911-093-6 | |  |
|  |  |  | |  |

A. To evaluate the material, a four-scale with additional motivating comments will be used.

A detailed report must be provided, clearly reflecting all strong and weak points.

The following rating scale applies:

1. Poor or no criteria covered. i.e. the material would require extensive changes to be made to be considered suitable.
2. Insufficiently covers the criteria.
3. Good coverage of criteria, with minor shortcomings.
4. Excellent, fully covers the criteria/suitable for its purpose.

B. The screener’s report MUST be completed in English (including reports on titles written in Afrikaans etc.)

|  |  |
| --- | --- |
| Each item scored out of 4 |  |
| SECTION 1: CONTENT/CONTEXT | 3 |
| SECTION 2: LEARNING ACTIVITIES AND ASSESSMENT | 2 |
| SECTION 3: LAYOUT, DESIGN AND OVERALL QUALITY | 4 |
| SECTION 4: LECTURER GUIDE | 3 |
| **TOTAL SCORE** | 12 |

|  |  |
| --- | --- |
| RECOMMENDED |  |
| CONDITIONALLY RECOMMENDED | x |
| NOT RECOMMENDED |  |

DEPARTMENT OF HIGHER EDUCATION AND TRAINING

**TVET COLLEGES NC(V)**

**STUDENT TEXTBOOKS AND LECTURER GUIDES**

**PUBLISHERS REPORT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | PUBLISHER | Future Managers | | |
| 2 | TITLE | NCV2 Basic Principles of Computer Programming and Computer Literacy | | |
| 3 | ISBN | 978-0-63911-092-9 | | |
| 4 | AUTHORS | Godwin Dzvapatsva | | |
| 5 | SUBJECT | Basic Principles of Computer Programming and Computer Literacy | LEVEL | 2 |

|  |  |
| --- | --- |
| RECOMMENDED |  |
| CONDITIONALLY RECOMMENDED | x |
| NOT RECOMMENDED |  |

**The author has written an excellent book that covers and explains the SAG content in a comprehensive manner**. It is possible to use the programming section of the textbook to facilitate a class lesson from the perspective of the teacher. In terms of the learner's perspective, it can be used for independent study.

This text is only conditionally approved due to the assessment activities and programming examples. In order to ensure the effectiveness of assessment activities, a more integrated approach is required that will test not only whether the student has mastered the basic concepts, but also whether the student is able to make the mental leap to realize that concepts previously learned need to be combined in order to successfully solve a problem.

As the student masters more concepts in each mode, the author may wish to build on a single example throughout and add additional features as necessary. Additionally, the author may link the Scratch examples to Python examples. Demonstrating how a scratch problem can now be solved using Python.

For assessment purposes, it would be beneficial for the author to include different types of programming questions. Typical examples include reading the code and predicting the output, completing the missing line of code, identifying the logic errors, etc. As a result, students will not be surprised when they encounter similar questions during an examination. It is also necessary to include some summative assessment activities where students write code in order to solve larger real-world problems.

**We commend your efforts and look forward to seeing how the changes are implemented.**

-The reviewers

|  |  |
| --- | --- |
| **STRONG POINTS** | **WEAK POINTS / SUGGESTIONS TOWARDS IMPROVEMENT** |
| Programming content goes into detail and explains concepts before showing practical application.  Introduced students to concept of interaction between Scratch and Pi SBC.  Content from different LO’s as specified int SAG is used to inform each other and not just discussed in isolation. Example is Turtle and Scratch colours in Module 3.  Module 4 on VCS is well written and technically excellent.  Using f string for string display. (Current recommended method) | Some of the content goes to far beyond the suggested LO’s in the SAG. The text can benefit from a slight streamline perhaps moving some content to companion text.  Needs more cross Topic integration for the Python programming. Many of the concepts is presented in isolation. But in the real world concepts will be combined to solve problems. Perhaps a few examples can be added where a problem is solved using different concepts.  The lecturer guide needs to provide possible lesson plans end schedules. Also need throughout assessment examples. |
| **Topic 1 (Module 1)**  Thank you for the development of a very detailed student text which includes a variety of elements to support the teaching and learning of the subject content.  The use of icons throughout the text is good to highlight and emphasise key points and concepts.  Overall Topic 1 is presented well. | Please include a single diagram such as this one to indicate the conversion from decimal to binary as well (page 4)  Decimal to Binary (Definition, Conversion, Table and Examples)  Correct the formatting error in the example 1.1 (answer)  Including an example in the following way could also support the mastering of the content  Binary, Hexadecimal, Decimal. Introduction to understanding binary… | by  Alisher Fayzimatov | Medium  Including an example in the following way could also support the mastering of the content.  The CPU cycle is not intrinsically part of the LOs and can be removed.  Section 1.28 and 1.29 requires a relook. The modular design refers to the composition and easily interchangeable components.  What are the components of a system unit?  Figure 1.15: Modules of a computer system as given in the text is misplaced and shows the architecture of the CPU??  The image is taken from Wikipedia and used out of context ????  <https://en.wikipedia.org/wiki/Computer_architecture>  Please add additional activities for LOs 1.3.5 to 1.3.15  Location and logic wise the binary conversion is well suited. But it is recommended to ease into the topic. Perhaps author can take a more gradual approach in introducing the topic.  Module text covers the content in great detail. It may be beneficial to slightly down the level of detail and shift some content to an appendix or companion chapter. Example of some instances where a content trim can be beneficial includes page 11 that mentions the different types of ROM page 12 that goes into very detailed explanation of hard disk drive and page 17 detailing the fetch decode execute cycle.  1.3.4 page 24. It would be useful to have a visual illustration of the different path components |
| **Topic 2 (Module 2)** | What on earth is **calculational thinking??**  There is no such concept??  The correct term is – **Computational thinking??**  The SDLC is different from the PDLC. The PDLC concerns itself with the coding process and less of the analysis and maintenance  Phases Of Program Development Life Cycle  Step 1 – Program specification  Step 2 – Program design  Step 3 – Coding  Step 4 – Program testing  Step 5 – Documentation  Step 6 – Maintenance  Compilers  Enhance the examples listed for compilers as well. There are more well-known programming languages to include.    We don’t agree with the advantages and disadvantages listed.  See  <https://www.businesscompilerng.com/2022/02/advantages-and-disadvantages-of_21.html>  <https://buggyprogrammer.com/advantages-and-disadvantages-of-compiler-and-interpreter/>  2.1.4 – This section is not correctly discussed and should be enriched please see:  <https://business.tutsplus.com/tutorials/problem-solving-skills-that-lead-to-solutions--cms-39747>  2.1.6 – This section is discussed theoretically but should be enriched with some examples.  2.2.3 – This section presents the use of variables, yet variables have not been introduced yet. Please rework the section and include a short introduction to variables.  The students will also not know what a loop is.  Start off with simpler examples!  The initial examples in 2.2.3 is too complex!  **Section 2.2.9 should** be expanded with additional examples and should also include more exercises and activities!  Please create scratch equivalent examples of the following example programs as well.  Example 5.9 page 114  Example 5.14 page 125  Add a scratch program to convert binary do decimal and vice versa  Page 35 the difference between compiler and interpreter can perhaps be illustrated more effectively with an image or illustration.  2.1.5 page 37. The concept of Polya is very good. I may argue that the example problem being solved should be more relatable to the students then the current math example. Perhaps the author can illustrate how the 4 steps can be used in solving a real word problem.  Section 2.2.2 seems a bit contrived when tasking into consideration the level of student.  Section 2.2.6 page 45 again a real-world example for the IPO chart may be advisable. It is also advisable to expand the IPO chart section as this is directly mentioned in the SAG. |
| **Topic 3 (Module 3)** | Write equivalent scratch programs for the examples covered in module 2, e.g. 2.8, 2.9, 2.10 and 2.11. This will reinforce the concepts and content.  We do however require the inclusion of at least 2 more examples and at least SIX or so more practical activities and exercises! It is imperative that the students are provided with ample reinforcement activities.  For SO 3.1 the following LOs highlighted in RED should receive more attention in the text with additional reinforcement example’s  3.1.5 Demonstrate a knowledge of various concepts encapsulated in a block-based language such as:  3.1.5.1 Differentiating between a sprite and an image  3.1.5.2 Using existing images and sprites  3.1.5.3 Creating a user defined sprite  3.1.5.4 Different categories of block-based coding blocks (e.g. events, motion, control, variables…. Etc.)  3.1.5.5 Changing appearance and costumes  **3.1.5.6 Adding buttons**  3.1.5.7 Assigning events and triggers  3.1.5.8 Creating and using variables  3.1.5.9 Operations on data  3.1.5.10 Comparisons  3.1.5.11 Broadcasting  3.1.5.12 Repeating actions  3.1.5.13 Using conditional statements  **3.1.5.14 Manipulating strings**  **3.1.5.15 Basic list operations**  3.1.5.16 incorporating sounds and images  **3.1.5.17 incorporating multiple sprites**  More activities and exercises are required. S  -- Turtle Python section is well explained and introduced gradually.  3.3.6 page 91. Student encouraged to change line colour. Perhaps just give a few colour options to guide the student thinking into what is possible. |
| **Topic 4 (Module 4)** | Excellent module on VCS! |
| **Topics 5 to 11 (Modules 5 to 11)** | Well written in general.  The concept of showing the number conversions manual steps (module 6) and then showing the same conversion in Python is good. One does think the text can benefit from showing some larger examples that combine different elements of programming in a larger example.  Module 7 page 179. The f statement is introduced in combination with the date time module. Perhaps a more phased approach can be used. Start by doing string with basic example and build from there.  Showing the while loop combined with the if in Module 9 page 224 is a good start toward integrating different programming concepts into larger examples.  Sections 6.3, 6.4 and 6.5 needs to be expanded with the inclusion of examples and questions.  E.g., Section 6.5 covers string functions yet there are no integrated examples and exercises.  Please include some examples of where program output is written to a textile. E.g. Very simple invoice or report etc. |

##### 

##### GUIDING CRITERIA FOR EVALUATING OF STUDENT TEXTBOOKS AND LECTURER GUIDES

**TVET College Curriculum: National Certificate (Vocational)**

**GENERAL CRITERIA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SECTION 1: CONTENT / CONTEXT** | | | | |
|  | **Yes** | **No** | **Remarks** |  |
| 1.1 | ✓ |  | Some of the early Modules can be reworked slightly to reduce detail. Programming modules need not be reduced in content. | The LM content covers all the Learning Outcomes and the Assessment Standards of the subject. |
| 1.2 | ✓ |  |  | The LM is appropriately sequenced. |
| 1.3 |  |  | See comments above | The LOs are comprehensively covered according to the weighting. |
| 1.4 | ✓ |  | Requires more integrated programming examples where different concepts are combined to solve problems. | The content matter and examples are current and up-to-date. |
| 1.5 | ✓ |  |  | The content places learning in context i.e. integrates Assessment Standards within the subject to give students an authentic learning experience. |
| 1.6 | ✓ |  |  | There is clear integration of theory and applied competence. |
| 1.7 | ✓ |  |  | The content matter and examples/samples/visuals are sensitive to diversity e.g. culture, religion, gender, etc. |
| 1.8 | ✓ |  |  | The LM provides a variety of meaningful activities for individuals, pairs and groups. |
| 1.9 | ✓ |  | Generally, yes but see comments where content can be reduced or moved to appendix. | The level of the content is appropriate for the specific level. |
| 1.10 | ✓ |  |  | The language usage is simple, clear and understandable for students on the specific level. |
| 1.11 | ✓ |  |  | The language usage and style make the LM accessible for students by for example including features such as a dictionary or glossary of terms |
| 1.12 | ✓ |  |  | Key concepts and terms are clearly defined |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SECTION 2: LEARNING ACTIVITIES AND ASSESSMENT** | | | | | | |
|  | **Yes** | **No** | | **Remarks** | |  |
| 2.1 | ✓ |  | |  | | Assessment tasks are derived from LOs and Ass and are therefore appropriate. |
| 2.2 |  | × | | Larger more contrived exercises needs to be included that will test the combination of different programming concepts in larger problems. | | The assessment activities deliver sufficient, current, valid and authentic evidence for the students’ PoEs. |
| 2.3 | ✓ |  | |  | | Assessment tasks are aligned to the Programme of Assessment as described in the Assessment Guidelines. |
| 2.4 |  | × | | Can include a few different question types. | | A variety of assessment tasks are used. |
| 2.5 | ✓ |  | |  | | Assessment targets student achievement at different levels of complexity. |
| 2.6 | ✓ |  | |  | | Assessment tasks are clearly formulated and unambiguous. |
| 2.7 | ✓ |  | |  | | Assessment tasks and learning activities provide for regular assessment. |
| 2.8 | ✓ |  | |  | | Assessment activities reflect the integration of Assessment Standards. |
| SECTION 3: LAYOUT, DESIGN AND OVERALL QUALITY | | | | | | |
|  | **Yes** | | **No** | **Remarks** |  | |
| 3.1 | ✓ | |  |  | The text is structured, using headings and subheadings. | |
| 3.2 | ✓ | |  |  | The font and typeface are clear and easy to read. | |
| 3.3 | ✓ | |  |  | The illustrations and diagrams are clear and relevant, without bias. | |
| 3.4 | ✓ | |  |  | The paper is of a good quality and bound securely. | |
| 3.5 | ✓ | |  |  | The LM has a contents page with clear reference to chapters and page numbers. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SECTION 4: LECTURER GUIDE (where developed) | | | | |
|  | **Yes** | **No** | **Remarks** |  |
| 4.1 | ✓ |  |  | Provides clear and systematic guidance on the use of the LM. |
| 4.2 |  | - |  | Provides examples of a work schedule that speaks to the activities in the text. |
| 4.3 | ✓ |  |  | Includes exemplar assessments for the level that speak to the learning tasks in the text. |
| 4.4 | ✓ |  |  | Provides suggested answers/solutions, memoranda, checklists, rubrics, etc. that match the assessment tasks in the LM. |