REPUBLIC OF KENYA

COMPETENCY-BASED MODULAR CURRICULUM

FOR

SOFTWARE DEVELOPMENT

KNQF LEVEL 6

CYCLE 3

PROGRAMME ISCEDCODE: 0613 554 A

TVET CDACC
P.O. BOX 15745-00100
NAIROBI

FOREWORD

The provision of quality education and training is fundamental to the Government's overall strategy for socio-economic development. Quality education and training contribute to achievement focused on Kenya's development blueprint and sustainable development goals.

Reforms in the education and training sector are necessary for achievement of Kenya Vision 2030 and meeting the provisions the Constitution of Kenya. The education sector had to be aligned to the Constitution and this resulted in formulation of the Policy Framework for Reforming Education and Training (Sessional Paper No. 1 of 2019). A key feature of this policy is the change in the design and delivery of TVET training. The reforms include making TVET competency-based, developing the curriculum in collaboration with industry, certifying learners based on demonstrated competence, and allowing multiple entry and exit points in TVET programmes.

These reforms emphasize the role of industry as key collaborators in curriculum development to ensure it aligns with their competence needs. It is against this background that this Curriculum has been developed.

It is my conviction that this curriculum will play a great role towards development of competent human resource for the ICT sector's growth and sustainable development.

PRINCIPAL SECRETARY
STATE DEPARTMENT FOR TVET
MINISTRY OF EDUCATION

PREFACE

Kenya Vision 2030 aims to transform Kenya into a newly industrializing middle-income country, providing high-quality life to all its citizens by the year 2030. Kenya intends to create globally competitive and adaptive human resource base to meet the requirements of a rapidly industrializing economy through lifelong education and training. TVET has a responsibility to facilitate the process of inculcating knowledge, skills, and worker behaviour necessary for catapulting the nation to a globally competitive country, hence the paradigm shift to embrace Competency-Based Education and Training (CBET).

TVET Act, CAP 210A and Sessional Paper No. 1 of 2019 on Reforming Education and Training in Kenya for Sustainable Development emphasized the need to reform curriculum development, assessment, and certification. This called for a shift to CBET to address the mismatch between skills acquired through training and skills needed by industry, as well as increase the global competitiveness of the Kenyan labour force.

This curriculum has been developed in adherence to the Kenya National Qualifications Framework and CBETA standards and guidelines. The curriculum is designed and organized into Units of Learning with Learning Outcomes, suggested delivery methods, learning resources, and methods of assessing the trainee's achievement. In addition, the units of learning have been grouped in modules to concretize the skills acquisition process and streamline upskilling.

I am grateful to all expert trainers and everyone who played a role in translating the Occupational Standards into this competency-based modular curriculum.

COUNCIL CHAIRPERSON
TVET CDACC

ACKNOWLEDGEMENT

This curriculum has been designed for competency-based training and has independent units of learning that allow the trainee flexibility in entry and exit. In developing the curriculum, significant involvement and support were received from expert trainers, institutions and organizations.

I recognize with appreciation the role of the ICT National Sector Skills Committee (NSSC) in ensuring that competencies required by the industry are addressed in the curriculum. I also thank all stakeholders in the ICT sector for their valuable input and everyone who participated in developing this curriculum.

I am convinced that this curriculum will go a long way in ensuring that individuals aspiring to work in the ICT Sector acquire competencies to perform their work more efficiently and effectively.

COUNCIL SECRETARY/CEO
TVET CDACC

TABLE OF CONTENTS

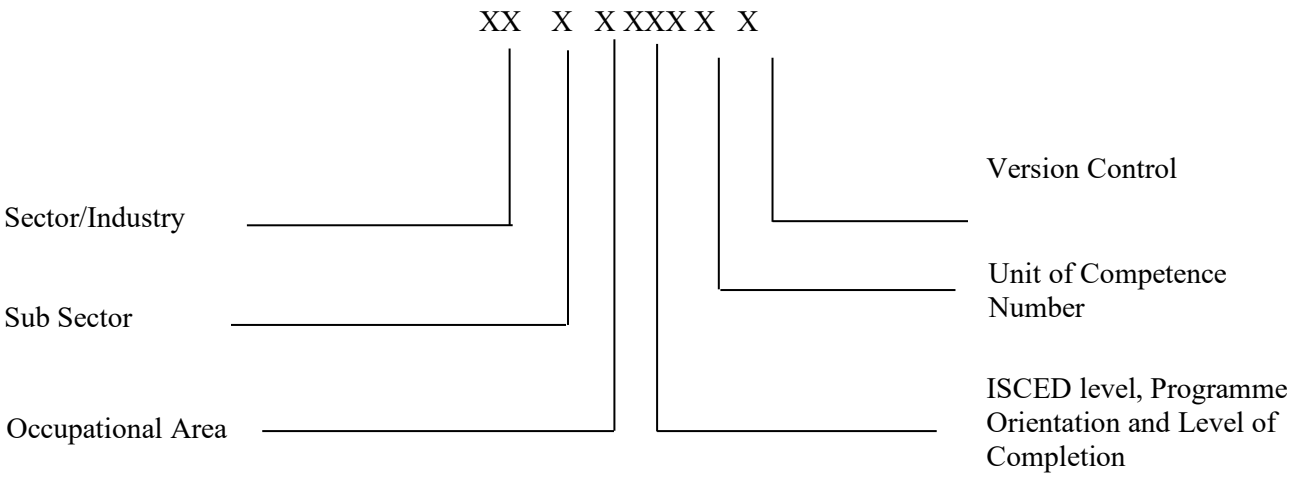
FOREWORD 1
ACKNOWLEDGEMENT 3
TABLE OF CONTENTS 4
ACRONYMS AND ABBREVIATIONS 5
KEY TO UNIT CODE 7
COURSE OVERVIEW 8
MODULE I 13
COMPUTER OPERATIONS 14
STRUCTURED PROGRAMMING 25
SOFTWARE SYSTEM REQUIREMENTS 31
MODULE II 37
DISCRETE MATHEMATICAL CONCEPTS 38
APPLICATION END-USER SUPPORT 45
MODULE III 50
ENTREPRENEURIAL SKILLS 51
COMPUTERISED DATABASE SYSTEMS MANAGEMENT 1
MODULE IV 9
PROJECT MANAGEMENT PRINCIPLES 10
WORK ETHICS AND PRACTICES 18
WEB APPLICATION DEVELOPMENT 23
MODULE V 31
OBJECT ORIENTED PROGRAMMING 32
DESKTOP APPLICATION DEVELOPMENT 42
MODULE VI 50
MOBILE APPLICATION DEVELOPMENT 51
COMMUNICATION SKILLS 58

ACRONYMS AND ABBREVIATIONS

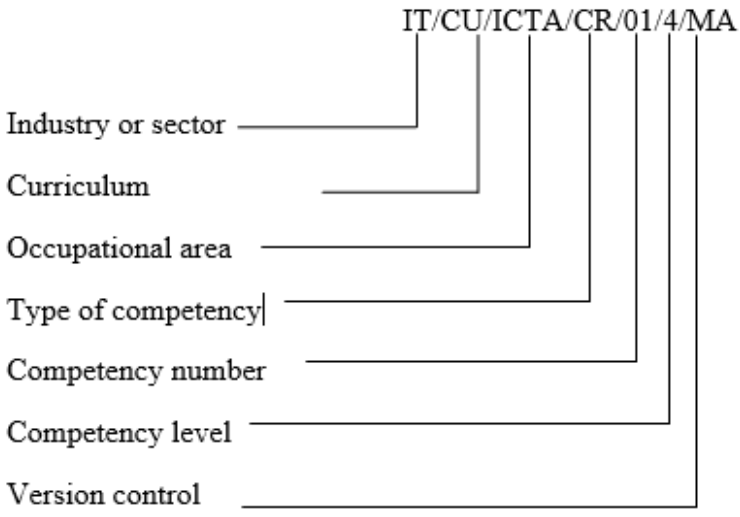
Acronym/Abbreviation	Description
API	Application Programming Interface
ASCII	American Standard Code Information Interchange
BCD	Binary Coded Decimal
CBET	Competency Based Education and Training
CSS	Cascading Style Sheet
DBMS	Database Management System
DNS	Domain Name Service
FTP	File Transfer Protocol
GCE	General Certificate of Education
GIT	Global Information Tracker
HDD	Hard Disk Drive
HTML	Hypertext Mark-up Language
ICT	Information Communication Technology
IDE	Integrated Development Environment
ISCED	International Standard Classification of Education
ISO	International Organization For Standardization
KACE	Kenya Advanced Certificate of Education
KCE	Kenya Certificate of Education
KCSE	Kenya Certificate of Secondary Education
KNQF	Kenya National Qualification Framework
MVC	Model View Controller
OOP	Object Oriented Programming
OSHA	Occupation Safety and Health Administration
PDF	Portable Document Format
PERT	Program Evaluation Review Techniques
PHP	Hypertext Pre-Processor
SDK	Software Development Kit

SMART	Specific Measurable Achievable Realistic Time-Bound
SQL	Structured Query Language
SSD	Solid State Disk
SSL	Secure Socket Layer
TLS	Transport Layer Security
TV	Television
TVET	Technical Vocational Education and Training
TVETA	Technical and Vocational Education Training Authority
UX	User experience
WBS	Work Breakdown Structure

KEY TO UNIT CODE



KEY TO TVET CDACC UNIT CODE



COURSE OVERVIEW

Description of the Course

The Software Developer Level 6 curriculum prepares learners with the technical skills and knowledge needed to design, develop, and maintain software applications. It comprises of basic learning in communication skills, work ethics and entrepreneurial skills. In additional, this curriculum entails the following foundational common units; computer operations, discrete mathematics and project management. Core units of learning include advanced computerized database systems, computerized database systems management, software requirements analysis, web application development, object oriented programming, desktop application, mobile application development and application end-user support. The program emphasizes practical experience through 480 hours of industry training, bridging the gap between classroom learning and industry demands. Graduates are equipped for careers in software development, web and mobile applications, database management, and IT support, making them ready to thrive in the dynamic ICT sector.

Summary of Units of Learning

ISCED Unit Code	TVET CDACC UNIT CODE	Unit Title	Unit Duration (Hours)	Credit Factor
MODULE I				
0611 441 0	ICT/CU/SD/CC/01/5/MA	COMPUTER OPERATIONS	90	9.0
0613 451 0	ICT/CU/SD/CC/02/5/MA	STRUCTURED PROGRAMMING	240	24.0
0611 451 0	ICT/CU/SD/CR/01/5/MA	SOFTWARE SYSTEM REQUIREMENTS	120	12.0
MODULE II				

0541 451 0	ICT/CU/SD/CC/03/5/MA	DISCRETE MATHEMATICAL CONCEPTS	160	16.0
0611 451 0	ICT/CU/SD/CR/02/5/MA	APPLICATION END-USER SUPPORT	150	15.0
MODULE III				
0413 441 0	ICT/CU/SD/BC/01/5/MA	ENTREPRENEURIAL SKILLS	40	4.0
0612 451 0	ICT/CU/SD/CR/04/5/MA	COMPUTERIZED DATABASE SYSTEMS	250	25.0
MODULE IV				
0688 451 0	ICT/CU/SD/CC/04/5/MA	PROJECT MANAGEMENT PRINCIPLES	110	11.0
0417 441 0	ICT/CU/SD/BC/02/5/MA	WORK ETHICS AND PRACTICES	40	4.0
0417 441 0	ICT/CU/SD/CR/05/5/MA	WEB APPLICATION DEVELOPMENT	250	25.0
MODULE V				
0613 551 0A	ICT/CU/SD/CR/01/6/MA	OBJECT ORIENTED PROGRAMMING	170	17.0
0613 551 0	ICT/CU/SD/CR/02/6/MA	DESKTOP APPLICATION DEVELOPMENT	240	24.0
MODULE VI				

0613 551 0	ICT/CU/SD/CR/03/6/MA	MOBILE APPLICATION DEVELOPMENT	270	27.0
0031 541 0	ICT/CU/SD/BC/01/6/MA	COMMUNICATION SKILLS	40	4.0
Sub Total			2230	223.0
Industry Training			480	48.0
GRAND TOTAL			**2710**	**271.0**

Entry Requirements

An individual enrolling for this course should have any of the following minimum requirements:

a) Kenya Certificate of Secondary Education (K.C.S.E.) with a minimum mean grade of C- (minus)

b) Software Developer KNQF level 5 Certificate or its equivalent as determined by TVETA.

Trainer Qualification

A trainer for any of the units of competency in this course must:

a) Have at minimum a KNQF Level 7 qualification or its equivalent in a trade area related to this course.

b) Be registered by TVETA.

Industry Training

An individual enrolled in this course will be required to undergo Industry training for a minimum period of 480 hours in ICT sector. The industrial training may be taken after completion of all units for those pursuing the full qualification or be distributed equally in each unit for those pursuing part qualification. In the case of dual training model, industrial training shall be as guided by the dual training policy.

Assessment

The course shall be assessed formatively and summatively:

a) During formative assessment all performance criteria shall be assessed based on performance criteria weighting.

b) Number of formative assessments shall minimally be equal to the number of elements in a unit of competency.

c) During summative assessment basic and common units may be integrated in the core units or assessed as discrete units.

d) Theoretical and practical weighting for each unit of learning shall be as follows;

   i) 30:70 for units in modules I, II, III AND IV

   ii) 40:60 for units in modules V and VI

e) Formative and summative assessments shall be weighted at 60% and 40% respectively in the overall unit of learning score

   For a candidate to be declared competent in a unit of competency, the candidate must meet the following conditions:

   i) Obtained at least 40% in theory assessment in formative and summative assessments.

   ii) Obtained at least 60% in practical assessment in formative and summative assessment where applicable.

   iii) Obtained at least 50% in the weighted results between formative assessment and summative assessment where the former constitutes 60% and the latter 40% of the overall score.

f) Assessment performance rating for each unit of competency shall be as follows:

MARKS	COMPETENCE RATING
80 -100	Attained Mastery
65 - 79	Proficient
50 - 64	Competent
49 and below	Not Yet Competent
Y	Assessment Malpractice/irregularities

g) Assessment for Recognition of Prior Learning (RPL) may lead to award of part and/or full qualification.

Certification

A candidate will be issued with a Certificate of Competency upon demonstration of competence in a core Unit of Competency. To be issued with Kenya National TVET Certificate in Software Development KNQF Level 6 the candidate must demonstrate competence in all the Units of Competency as given in the qualification pack. Statement of Attainment certificate may be issued upon demonstration of competence in a certifiable element within a unit.

The certificates will be issued by TVET CDACC

MODULE I

Unit Category	**Unit Code**	**TVET CDACC UNIT CODE**	**Unit Name**	**Unit Duration (Hours)**
COMMON	0611 441 03A	ICT/CU/SD/CC/01/5/MA	COMPUTER OPERATIONS	90
COMMON	0613 451 06A	ICT/CU/SD/CC/02/5/MA	STRUCTURED PROGRAMMING	220
CORE	0611 451 07A	ICT/CU/SD/CR/01/5/MA	SOFTWARE SYSTEM REQUIREMENTS	110
			TOTAL	**420**

COMPUTER OPERATIONS

ISCED UNIT CODE: 0611 441 03A

TVET CDACC UNIT CODE: ICT/CU/SD/CC/01/5/MA

Duration of Unit: 90 hours

Relationship to Occupational Standards

This unit addresses the Unit of Competency: Apply Computer Operations

Unit Description

This unit covers the competencies required to apply computer operations. It involves processing computerized word documents, manipulating computerized spreadsheets, maintaining computerized databases, prepare presentation slides, manipulating graphic application and performing online collaboration.

Summary of Learning Outcomes

Learning Outcomes	Duration (Hours)
1. Process computerized word document	20
2. Manipulate computerized spreadsheet	25
3. Maintain computerized database	15
4. Prepare Power point presentation	10
5. Manipulate graphic application	10
6. Perform online collaboration	10
TOTAL	**90**

Learning Outcomes, Content and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Process computerized word document	1.1 Ergonomic risk factors 1.2 Operating Computer devices 1.2.1 Meaning and importance of computer	● Practical assessment ● Simulations ● Project ● Observation Checklist

	1.2.2 Functions and Uses of Computers 1.2.3 Classification of computers 1.2.4 Components of a computer system 1.2.5 Computer Hardware 1.2.5.1 The System Unit 1.2.5.2 Input Devices 1.2.5.2.1 Mouse use techniques 1.2.5.2.2 Keyboard Parts and Use Techniques 1.2.5.3 Output Devices 1.2.5.4 Storage Devices 1.2.5.5 Computer Ports 1.2.6 Procedure for turning/off a computer 1.2.7 Desktop Customization 1.2.8 File and Files Management using an operating system 1.2.9 Computer external devices management 1.3 Creation of computerized word document 1.3.1 Introduction to word document 1.3.2 Types of word processors 1.3.3 Creating word document 1.4 Editing and formatting word document 1.3.4 Word document editing features 1.3.4.1 Text editing 1.3.4.2 Paragraph editing 1.3.4.3 Document editing 1.4.1 Word document formatting features 1.3.4.4 Text formatting 1.3.4.5 Paragraph formatting 1.3.4.6 Document formatting	• Product Checklist • Written assessment • Portfolio of evidence

	1.4.2 Enhancing productivity 1.3.4.7 Set basic options/ preferences 1.3.4.8 Help resources 1.3.4.9 Use magnification/zoom tools 1.3.4.10 Display, hide built-in tool bar 1.5 Mail merge 1.5.1 Mail merge preparation 1.5.2 Mail merge output 1.6 Printing of computerized word document 1.6.1 Print setup 1.6.2 Printing	
2. Manipulate computerized spreadsheet	2.1 Creation of Computerized spreadsheet workbook 2.1.1 Spreadsheet concepts 2.1.2 Elements of spreadsheet window 2.1.2.1 Worksheet 2.1.2.2 workbook 2.1.2.3 Rows 2.1.2.4 columns 2.1.2.5 Cells 2.2 Cell referencing 2.2.1.1 Relative cell referencing 2.2.1.2 Absolute cell referencing 2.2.1.3 Mixed cell referencing 2.2.2 Spreadsheet editing features 2.2.2.1 Worksheet editing 2.2.2.2 Inserting	• Practical assessment • Simulations • Project • Observation Checklist • Product Checklist • Written assessment • Portfolio of evidence

	rows/columns 2.2.2.3 Removing rows/columns 2.2.2.4 Adjusting row heights and column width 2.2.2.5 Inserting worksheets 2.2.2.6 Renaming worksheets 2.2.2.7 Move or copy worksheets 2.2.2.8 Deleting worksheets 2.2.3 Data manipulation in spreadsheets 2.2.3.1 Data entry 2.2.3.2 Types of data 2.3 Formulas and functions 2.3.1.1 Formulas and functions syntax 2.3.1.2 Arithmetic functions 2.3.1.3 logical functions 2.3.1.4 Look up functions 2.3.2 Computerized spreadsheet worksheet formatting 2.3.2.1 Font styles 2.3.2.2 Alignment 2.3.2.3 Borders and shading 2.3.2.4 Header and footer 2.4 Charts generation 2.4.1.1 Types of charts 2.4.1.2 Insert charts 2.4.1.3 Labelling and Editing charts 2.4.1.4 Computerized spreadsheet	

	workbook printing 2.4.1.5 Print setup 2.4.1.6 Printing	
3. Maintain computerised database	3.1 Computerised database user requirements collection 3.1.1 Introduction to database 3.1.1.1 Key concepts 3.1.1.2 Database organisation 3.1.1.3 Database relationships 3.1.1.4 Database operations 3.1.2 Collection of User requirements 3.2 Design Computerised database schema 3.2.1 Creating database models 3.2.1.1 ERD models 3.2.1.2 Relational models 3.3 Creation of Computerised database objects 3.3.1 Database Objects 3.3.1.1 Tables 3.3.1.2 Records 3.3.1.3 Fields 3.3.1.4 Keys 3.3.1.5 Forms 3.3.1.6 Queries 3.3.1.7 Reports 3.4 Data manipulation 3.4.1 Inserting records 3.4.2 Retrieving records 3.4.3 Deleting records 3.4.4 Updating record 3.4.5 Printing database objects 3.4.5.1 Tables 3.4.5.2 Forms	• Practical assessment • Simulations • Project • Observation Checklist • Product Checklist • Written assessment • Portfolio of evidence

	3.4.5.3 Queries 3.4.5.4 Reports	
4. Prepare Power point presentation	4.1 Collection of Presentation requirements 4.1.1 Definition of terms 4.2.1 Presentation requirements 4.2.2 Types of presentation software 4.2.3 Elements of presentation window 4.2.4 Manipulating presentations 4.2.4.1 Create a PowerPoint presentation 4.2.4.2 Save a PowerPoint presentation 4.2.5 Working with presentations 4.2.5.1 Switch between open PowerPoint presentations 4.2 Design presentation layout 4.2.1 Types of presentation layout 4.2.2 Factors to consider when designing presentation layout 4.3 Creation of a Slide 4.3.1 Slide views 4.3.2 Slide designs 4.3.3 Slide transition 4.4 Manipulation of a slide 4.4.1 Adding data/text to a slide 4.4.2 Slide animation 4.4.3 Formatting data/text 4.4.4 Move/copy/delete a slide 4.4.5 Inserting header and footer 4.4.6 Presentation objects	• Practical assessment • Simulations • Project • Observation Checklist • Product Checklist • Written assessment • Portfolio of evidence

	4.2.5.2 Tables 4.2.5.3 charts 4.4.7 Print setup 4.2.5.4 Printing PowerPoint presentation	
5. Manipulate graphic application	5.1 Collecting graphic design requirements 5.1.1 Definition of terms 5.1.2 Graphic application requirements 5.1.3 Types of graphic application software 5.1.4 Types of publications designs 5.1.4.1 Templates 5.1.4.2 Banners 5.1.4.3 Booklets 5.1.4.4 Brochures 5.1.4.5 Flyers 5.1.4.6 Posters 5.1.4.7 Cards 5.1.4.8 Certificates 5.1.4.9 Magazines 5.1.5 Elements of Graphic application window 5.2 Creation of graphic design 5.2.1 Perform basic tasks using graphic application software 5.2.1.1 Publication type 5.2.1.2 Page setup 5.2.1.3 Ruler/guides 5.2.1.4 Page views 5.2.2 Add content to a publication 5.2.3 Edit content to a publication 5.2.4 Format text and paragraphs in a publication	• Practical assessment • Simulations • Project • Written assessment • Portfolio of evidence

	5.2.5 Page formatting in a publication 5.2.5.1 Columns 5.2.5.2 Borders and shading 5.2.5.3 Headers and footers 5.2.5.4 Background 5.2.5.5 Watermarks 5.2.5.6 Orientation 5.2.6 Work with graphics objects in a publication 5.2.6.1 Textbox 5.2.6.2 Tables 5.2.6.3 Shapes 5.2.6.4 Pictures 5.2.6.5 (PNG, JPEG, GIF) 5.3 Publishing of graphic design 5.3.1 Prepare a publication 5.3.2 Print setup 5.3.3 Printing publication	
6. Perform Online Collaboration	6.1 Identification of Online collaboration tools 6.1.1 Definition of online collaboration 6.1.2 Importance of online collaboration 6.1.3 Factors to consider when choosing an online collaboration tool 6.1.4 Online collaboration tools 6.1.4.1 Microsoft teams 6.1.4.2 Skype 6.1.4.3 Google drive 6.1.4.4 Zoom 6.1.4.5 Google meet 6.1.4.6 Slack 6.2 Online collaboration preparation	• Practical assessment • Simulations • Project • Observation Checklist • Product Checklist • Written assessment • Portfolio of evidence

	6.2.1 Online collaboration key concepts 6.2.2 Common setup features 6.2.2.1 Download software to support online collaboration tools 6.2.2.2 Register and/ or set a user account 6.2.3 Preparation for online collaboration 6.3 Application of online collaborative tools 6.3.1 Using online collaborative tools 6.3.1.1 Online storage media 6.3.1.2 Using email 6.3.1.2.1 Sending and receiving email 6.3.1.2.2 Tools and settings 6.3.1.2.3 Organizing email 6.3.1.3 Using calendars 6.3.1.4 Online calendars 6.3.1.5 Social media 6.3.1.6 Online learning environment 6.3.1.7 Synchronization tools 6.4 Demonstrating Mobile collaborations 6.4.1 Key concepts in mobile applications 6.4.2 Mobile applications permissions 6.4.3 Synchronization	

Suggested Delivery Methods

- Demonstration by trainer
- Practical work by trainee
- Viewing of related videos
- Group discussions
- Facilitation using active learning strategies

Recommended Resources for 25 Trainees

S/No.	Category/Item	Description/ Specifications	Quantity	Recommended Ratio (Trainee: Item)
A	**Learning Materials**			
1.	Textbooks		5 pcs	5:1
2.	Installation manuals		5 pcs	5:1
3.	Flip Charts		5 pcs	5:1
4.	PowerPoint presentations	For trainer's use		
5.	Magazines/brochures/business cards			
B	**Learning Facilities & infrastructure**			
6.	Lecture/theory room		1	25:1
7.	Laboratory		1	25:1
C	**Consumable materials**			
8.	Printing papers		1 ream	1:20
9.	Foolscaps		1 ream	
10.	Toners/cartridges		2 pcs	13:1
11.	Assorted colour of			

	whiteboard markers			
D	**Tools and Equipment**			
12.	Computers		25 pcs	1:1
13.	Projector		1 pc	25:1
14.	Printers		2 pcs	1:13
15.	Whiteboard		1 pc	25:1
16.	Flash drives		5 pcs	5:1
17.	1 External Hard drive		1 pcs	25:1
18.	Application software suite		5 pcs	5:1

STRUCTURED PROGRAMMING

ISCED UNIT CODE: 0613 451 06A

TVET CDACC UNIT CODE: ICT/CU/SD/CC/02/5/MA

Relationship to Occupational Standards

This unit addresses the Unit of Competency: Apply Structured Programming

Duration of Unit: 220 Hours

Unit Description

This unit covers the competencies required to apply structured programming. It involves applying computer programming basics, writing program code, implementing program logic and implementing modular programming.

Summary of Learning Outcomes

Learning Outcomes	Duration (Hours)
1. Apply Computer Programming basics	30
2. Write program Code	40
3. Implement Program logic	70
4. Implement modular programming	80
TOTAL	**220**

Learning Outcomes, Content and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Apply Computer Programming basics	1.1. Programming Language Types 1.1.1. Overview of programming language categories 1.1.2. procedural 1.1.3. object-oriented	● Practical Activities ● Project work ● Group work ● Observation ● Portfolio of Evidence

	1.1.4. functional 1.2. Programming Paradigms 1.2.1. Explanation of common programming paradigms (e.g., imperative, declarative) 1.2.2. Choosing the appropriate paradigm based on project needs 1.3. Program Development Life Cycle 1.3.1. Stages of the program development life cycle (e.g., planning, design, implementation) 1.3.2. Best practices for adapting the life cycle to work requirements 1.3.3. Overview of program design tools (e.g. algorithms, flowcharts, wireframes, pseudocodes, decision table/trees) 1.4. Selecting design tools based on user requirements and project complexity	● Written tests
2. Write program Code	2.1. Program Writing Tools 2.1.1. Common program writing tools and IDEs (e.g. codeblocks, Turbo C, Eclipse) 2.1.2. Evaluating tools based on system requirements and developer preferences 2.2. Declaring Identifiers 2.2.1. Types of identifiers 2.2.1.1. Variable, 2.2.1.2. Functions 2.2.1.3. Arrays. 2.2.2. Ensuring identifiers align with program design specifications 2.2.3. Creating a naming convention guide for different types of identifiers.	● Practical Activities ● Project work ● Group work ● Observation ● Portfolio of Evidence ● Written tests

	2.2.4. Evaluating identifier 2.3. Initializing Variables and Constants 2.3.1. Importance of proper initialization in programming 2.3.2. Techniques for initialization based on design specifications 2.3.3. Writing code snippets demonstrating correct and incorrect initialization. 2.3.4. Best coding practices 2.3.4.1. Creating Comments in a program 2.3.4.2. Indenting statement 2.3.4.3. Program blocks of code 2.3.5. Conducting a workshop on variable and constant initialization techniques.	
3. Implement Program logic	3.1. Application of Data types 3.1.1. Data types 3.2. Application of program data control structures 3.2.1. Loops 3.2.1.1. For loops 3.2.1.2. While loops 3.2.1.3. Do while loops 3.2.2. Conditionals statements 3.2.2.1. If statements 3.2.2.2. Case statements 3.2.3. Best practices for implementing control structures as per design requirements 3.2.4. Solving coding challenges that require the use of different control structures. **3.2.5.** Creating flowcharts to visually represent control structures in a program. 3.3. Application of Data Structures 3.3.1. Overview of common data structures (e.g., arrays, linked lists) 3.3.2. Selecting appropriate data structures based on design specifications	• Practical Activities • Project work • Group work • Observation • Portfolio of Evidence • Written tests

	3.3.3. Implementing various data structures in a programming language of choice. 3.3.4. Comparing performance metrics of different data structures in a small project.	
4. Implement modular programming	4.1. Creation of Subroutines 4.1.1. Benefits of using subroutines (e.g., modularity, reusability) 4.1.2. Designing subroutines to meet user needs 4.1.3. Functions and subprograms 4.1.3.1. In built functions 4.1.3.2. User defined functions 4.1.3.2.1. Function parameters 4.1.3.2.2. Function return types 4.1.4. Design and implement a subroutine library for common tasks. 4.1.5. Program to create subroutines based on given specifications. 4.2. Application of data structures in looping through arrays in a function 4.2.1. Using various looping control structures 4.2.1.1. For loop 4.2.1.2. While 4.2.1.3. Do .. while 4.3. Perform debugging 4.3.1. Common debugging techniques and tools 4.4. Compiling a program 4.4.1. Compiling a series of programs with intentional error to learn about error messages.	● Practical Activities ● Project work ● Group work ● Observation ● Portfolio of Evidence ● Written tests

Suggested Delivery Methods

- Instructor led facilitation using active learning strategies
- Demonstration by trainer
- Practical work by trainees
- Viewing of related videos
- Field Visits
- Group Work
- Role plays
- Group projects

Recommended Resources for 25 Trainees

S/No.	Category/Item	Description/ Specifications	Quantity	Recommended Ratio (Item: Trainee)
A	**Learning Materials**			
1.	Internet connection	✓ For each computer	1	1:1
2.	Flip charts	A1	1	1:25
3.	Textbooks	For reference	3	3:25
B	**Learning Facilities & infrastructure**			
4.	Computer Laboratory	To accommodate 25 Learners	1	1:25
5.	Theory Room	furnished with 25 Arm-chairs and a suitable trainer's table	1	1:25
C	**Consumable materials**			
6.	Printing papers	A4	2 Reams	1:12
7.	Toner / Ink bottles	For printers	2 pcs	1:12
8.	White board markers	Assorted colors	20	4:5
D	**Tools and Equipment**			
9.	Computers	✓ Genuine Windows/Linux ✓ Genuine Microsoft office Software	25	1:1

		✓ Google Workspace Account ✓ Antivirus Software ✓ Suitable IDE		
10.	External storage media	HDD / SSD / Flash	1	1:25
11.	Printer	Working printer	2	1:12
12.	1 Smart-board / Smart TV / Projector (with screen)	Where available	1	1:25
13.	Whiteboard/Chalkboard	4 X 8 Feet	1	1:25

SOFTWARE SYSTEM REQUIREMENTS

ISCED UNIT CODE: 0611 451 07A

TVET CDACC UNIT CODE: ICT/CU/SD/CR/01/5/MA

Duration of Unit: 110 Hours

Relationship to Occupational Standards

This unit addresses the Unit of Competency: Establish Software System Requirements

UNIT DESCRIPTION

This unit covers the competencies required to establish software system requirements. It involves gathering user requirements, analyzing user requirements, and planning application deliverables.

Summary of Learning Outcomes

1. To gather user requirements
2. To analyze user requirements
3. To plan application deliverables

Learning Outcomes	Duration (Hours)
1. Gather user requirements	20
2. Analyze user requirements	40
3. Plan application deliverables	50
TOTAL	**110**

Learning Outcomes, Content and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Gather user requirements	1.1. Key project stakeholders 1.1.1. Project team 1.1.1.1. Team members & their roles 1.1.1.2. Project managers 1.1.2 System users 1.1.1.3. Developers 1.1.1.4. System Analysts 1.1.1.5. End-users 1.1.1.6. System Administrators 1.1.3 Organization management 1.1.1.7. Organization structure 1.2. Data collection tools 1.2.1 Methods of data collection 1.2.1.1. Questionnaires 1.2.1.2. Observations 1.2.1.3. Experimentation 1.2.1.4. Interviews 1.2.1.5. Surveys 1.2.1.6. Case Studies 1.2.3. Preparation of data collection tools 1.2.3.1. Questionnaires 1.2.3.2. Observations checklist 1.2.3.3. Interviews Questions	● Practical Activities ● Project work ● Demonstration ● Group Work ● Role Plays ● Observation ● Third Party report ● Portfolio of Evidence ● Written tests

	1.3. Collection of user requirements data: 1.3.1. Questionnaires 1.3.2. Observations 1.3.3. Experimentation 1.3.4. Interviews 1.3.5. Surveys 1.3.6. Case Studies	
2. Analyze user requirements	2.1. Types of user Requirements 2.1.1. Functional Requirements 2.1.2. Non-Functional Requirements 2.1.3. Technical Requirements 2.1.4. Budget and Timeline 2.1.5. Legal and Regulatory Requirements 2.2. Identification of System application requirements. 2.2.1. Purpose and Goals 2.3. Analysis of system application requirements 2.3.1. Feasibility Analysis: 2.3.2. Requirements Prioritization 2.3.3. Functional Analysis 2.3.4. Non-functional Analysis 2.3.5. Design Analysis 2.3.6. Risk Analysis 2.4. Documenting system application requirements specifications process	● Practical Activities ● Project work ● Demonstration ● Group Work ● Role Plays ● Observation ● Third Party report ● Portfolio of Evidence ● Written tests

	2.4.1. Functional Requirements 2.4.2. Non-Functional Requirements 2.4.3. Design Requirements 2.4.4. Technical Requirements 2.4.5. Assumptions and Constraints 2.4.6. Glossary 2.4.7. Appendices 2.4.8. Reviewing and revising system application requirement specifications.	
3. Plan application deliverables	3.1. System requirement Review Process 3.1.1. Review Criteria: 3.1.1.1. Traceability 3.1.1.2. Change Management 3.1.1.3. Version Control: 3.1.1.4. Testing and Validation 3.1.2. Legal Issues 3.1.2.1. Governing laws and international treaties 3.1.2.2. End-User License Agreement 3.1.3. Dispute resolution 3.1.4. Termination of Contracts 3.2. Creating a Project work plan 3.2.1. Project Scope 3.2.2. Key deliverables 3.2.2.1. Internal deliverables 3.2.2.2. External deliverables	• Practical Activities • Project work • Demonstration • Group Work • Role Plays • Observation • Third Party report • Portfolio of Evidence • Written tests

	3.2.3. Milestones 3.2.4. Timelines 3.2.5. Duties and responsibilities 3.2.6. Quality criteria definition 3.2.7. Constraints and dependencies 3.2.8. Resource sharing 3.2.9. Communication planning 3.3. Project development agreement 3.3.1. Contents 3.3.2. Importance 3.3.3. Sign-off	

Suggested Delivery Methods

- Instructor led facilitation using active learning strategies
- Demonstration by trainer
- Practical work by trainees
- Viewing of related videos
- Field Visits
- Group Work
- Role plays
- Group projects

Recommended Resources for 25 Trainees

S/No.	**Category/Item**	**Description/ Specifications**	**Quantity**	**Recommended Ratio** (Item: Trainee)
A	**Learning Materials**			
1	Internet connection	✓ For each computer	1	1:1
	Flip charts	A1	1	1:25
	Textbooks	For reference	3	3:25

B	**Learning Facilities & infrastructure**			
	Theory Room	furnished with 25 Arm-chairs and a suitable trainer's table	1	1:25
C	**Consumable materials**			
	Printing papers	A4	2 Reams	1:12
	White board markers	Assorted colors	20	4:5
D	**Tools and Equipment**			
	Computers	✓ Genuine Windows/Linux ✓ Genuine Microsoft office Software ✓ Google Workspace Account ✓ Antivirus Software ✓ MS Project / Ms Visio ✓ Visual Paradigm ✓ Clickup	25	1:1
	External storage media	HDD / SSD / Flash	1	1:25
	Printer	Working printer	2	1:12
	1 Smart-board / Smart TV / Projector (with screen)	Where available	1	1:25
	Whiteboard/Chalkboard	4 X 8 Feet	1	1:25

MODULE II

Unit Category	**Unit Code**	**TVET CDACC CODE**	**Unit Name**	**Unit Duration (Hours)**
COMMON	0541 451 04A		DISCRETE MATHEMATICAL CONCEPTS	160
CORE	0611 451 08A		APPLICATION END-USER SUPPORT	150
			TOTAL	**310**

DISCRETE MATHEMATICAL CONCEPTS

ISCED UNIT CODE: 0541 451 04A

TVET CDACC UNIT CODE: ICT/CU/SD/CC/03/5/MA

Relationship to Occupational Standards

This unit addresses the Unit of Competency: Apply Discrete Mathematical Concepts

Duration of Unit: 160 Hours

Unit Description

This unit covers the competence to apply discrete mathematical concepts. It involves carrying out set theory operations, performing matrix operations, applying number systems, applying logic gates, performing sequence and series operations, and demonstrating graph theory.

Summary of Learning Outcomes

Learning Outcomes	Duration (Hours)
1. Carry out set theory operations	32
2. Perform matrix operations	26
3. Apply number system	26
4. Apply logic gates	30
5. Perform sequence and series operations	20
6. Demonstrate graph theory	26
TOTAL	**160**

Learning Outcomes, Content and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Carry out set theory operations	1.1. Identify sets Characteristics 1.1.1. Definition 1.1.2. Order and Uniqueness 1.2. Methods of set representation 1.2.1. Roster Form 1.2.2. Set Builder Form 1.3. Cardinality of a set. 1.3.1. Finite 1.3.2. Infinite 1.4. Types of sets 1.4.1. Finite set 1.4.2. Infinite set 1.4.3. Empty set 1.4.4. Subset 1.4.5. Universal set 1.5. Venn Diagrams 1.5.1. Drawing Venn diagrams 1.6. Set Operations 1.6.1. Union 1.6.2. Intersection 1.6.3. Difference 1.6.4. Complement	• Practical Activities • Project work • Demonstration • Group Work • Observation • Portfolio of Evidence • Written tests
2. Perform matrix operations	2.1. Applying Matrix order 2.1.1. Dimension of matrix 2.1.2. Types of Matrices 2.1.2.1. Row matrix 2.1.2.2. Column matrix 2.1.2.3. Square matrix 2.1.2.4. Zero matrix 2.2. Matrix operations 2.2.1. Addition 2.2.2. Multiplication 2.2.3. Subtraction 2.3. Transpose of a matrix 2.3.1. Swapping rows and	• Practical Activities • Project work • Demonstration • Group Work • Observation • Third Party report • Portfolio of Evidence • Written tests

	columns 2.4. Transpose operations 2.4.1. Transpose 2.4.2. Sum 2.4.3. Product 2.5. Adjoint of a square matrix identification 2.6. Inverse of a square matrix identification. 2.6.1. Trace of a matrix application 2.6.2. Application of matrices 2.6.3. Computer Graphics 2.6.4. Statistics 2.6.5. Systems of Linear Equations	
3. Apply number Systems	3.1. Number systems 3.1.1. Definition of terms 3.1.2. Absolute values 3.1.3. Place values 3.1.4. Types of number systems 3.1.4.1. Decimal 3.1.4.2. Binary 3.1.4.3. Octal 3.1.4.4. Hexadecimal 3.2. Base conversion 3.2.1. Decimal to Other number system 3.2.2. Other number systems to decimal 3.2.3. Binary to other number systems 3.2.4. Other number systems to binary 3.3. Number systems arithmetic operations 3.3.1. Binary arithmetic 3.3.1.1. Addition, subtraction, multiplication and division	• Practical Activities • Project work • Demonstration • Group Work • Observation • Third Party report • Portfolio of Evidence • Written tests

	3.3.1.2. Ones and Twos complement 3.3.2. Octal arithmetic 3.3.2.1. Addition and subtraction 3.3.3. Hexadecimal arithmetic 3.3.3.1. Addition and subtraction 3.4. Binary codes 3.4.1. Binary coded decimal (BCD) 3.4.1.1. BCD operations 3.4.1.2. Addition and subtraction 3.4.2. ASCII 3.4.3. Gray Code 3.4.4. Excess-3	
4. Apply logic gates	4.1. Types of Logic gates 4.1.1. AND 4.1.2. OR 4.1.3. NOT 4.1.4. NAND 4.1.5. NOR 4.1.6. XOR 4.1.7. XNOR 4.2. Logic expressions 4.2.1. Logic circuit diagrams 4.2.2. Truth tables 4.3. Simplifying logic expressions 4.3.1. De-Morgan's theorems 4.3.2. Laws of Boolean algebra 4.3.2.1. Commutative 4.3.2.2. Associative 4.3.2.3. Distributive and more 4.3.2.4. Identity laws 4.3.2.5. Null laws 4.3.2.6. Complement laws 4.3.2.7. Commutative laws 4.3.3. Boolean expressions	• Practical Activities • Project work • Demonstration • Group Work • Observation • Third Party report • Portfolio of Evidence • Written tests

	simplification. 4.3.4. Application of Boolean Algebra. 4.3.5. Application of Karnaugh's Maps	
5. Perform sequence and series operations	5.1. Key terms of sequences. 5.1.1. Term 5.1.2. Index 5.1.3. General term (nth term) 5.1.4. Finite sequence 5.1.5. Infinite sequence 5.2. Summation of a sequence. 5.2.1. Arithmetic sum 5.3. Arithmetic series 5.3.1. General form of an arithmetic sequence 5.3.2. Sum of the first n terms 5.4. Geometric series 5.4.1. General form of a geometric sequence	• Practical Activities • Project work • Demonstration • Group Work • Observation • Third Party report • Portfolio of Evidence • Written tests
6. Demonstrate graph theory	6.1. Key Graph terminologies 6.1.1. Node 6.1.2. Edge 6.1.3. Vertex 6.1.4. Adjacent 6.2. Types of graphs 6.2.1. Null 6.2.2. Simple 6.2.3. Multigraph 6.2.4. Directed graphs 6.2.5. Undirected graphs 6.3. Representation of graphs 6.3.1. Adjacency Matrix 6.3.2. Adjacency List 6.3.3. Incidence Matrix 6.4. Application of graphs 6.4.1. Computer Networks 6.4.2. Social Networks 6.4.3. Transport Networks	• Practical Activities • Project work • Demonstration • Group Work • Observation • Third Party report • Portfolio of Evidence • Written tests

	6.4.4. Scheduling and Task Management	

Suggested Delivery Methods

- Instructor led facilitation using active learning strategies
- Demonstration by trainer
- Practical work by trainees
- Viewing of related videos
- Field Visits
- Group Work
- Role plays
- Group projects

Recommended Resources for 25 Trainees

S/No.	Category/Item	Description/ Specifications	Quantity	Recommended Ratio (Item: Trainee)
A	**Learning Materials**			
1.	Internet connection	✓ 5mbps	1	1:1
2.	Flip charts	A1	1	1:25
3.	Discrete Mathematics Textbooks	For reference	3	3:25
B	**Learning Facilities & infrastructure**			
4.	Theory Room	furnished with 25 Arm-chairs and a suitable trainer's table	1	1:25
C	**Consumable materials**			
5.	Printing papers	A4	5 Reams	1:25
6.	White board markers	Assorted colors	20	4:5
D	**Tools and Equipment**			
7.	External storage media	HDD / SSD / Flash	1	1:25
8.	Printer	Working printer	2	1:12

9.	1 Smart-board / Smart TV / Projector (with screen)	Where available	1	1:25
10.	Whiteboard/Chalkboard	4 X 8 Feet	1	1:25

APPLICATION END-USER SUPPORT

ISCED UNIT CODE: 0611 451 08A

TVET CDACC UNIT CODE: ICT/CU/SD/CR/03/5/MA

Relationship to Occupational Standards

This unit addresses the Unit of Competency: Offer Application End-User Support

Duration of Unit: 150 Hours

Unit Description

This unit covers the competencies required to offer Application End-user support. It involves developing application technical documents, performing application user training, gathering user feedback and performing application maintenance.

Summary of Learning Outcomes

Learning Outcomes	Duration (Hours)
1. To develop application technical documents	32
2. To perform application user training	42
3. To gather user feedback	33
4. To perform application maintenance	43
TOTAL	**150**

Learning Outcome	Content	Suggested Assessment Methods
1. Develop application technical documents	1.1 Identify types of application technical Documents 1.2 Prepare application technical documents 1.2.1 Contents / format & importance of each 1.2.2 Software requirement specification 1.2.3 Technical design documents 1.2.4 User interface design document 1.2.5 Database design document 1.2.6 Test plan and test case 1.2.7 Installation and deployment guide 1.2.8 User manual or user guide 1.2.9 API documentation 1.3 Validation of application technical documents	● Written tests ● Practical Activities ● Project work ● Third Party report ● Portfolio of Evidence
2. Perform application user training	2.1 Training needs assessment 2.1.1 Definition of TNA 2.1.2 Reasons for carrying out TNA 2.1.3 End User Training 2.1.4 Importance of end user training 2.1.5 User & customer training methods 2.1.5.1 Classroom mode / face-to-	● Practical Activities ● Project work ● Demonstration ● Group Work ● Role Plays ● Observation ● Third Party report ● Portfolio of Evidence

	face training 2.1.5.2 Automated online (virtual / Simulation) training 2.1.5.3 Self-paced learning modules 2.1.5.4 On-the-Job / On-Site training 2.2 Prepare end user training resources 2.2.1 Tutorials 2.2.2 Frequently asked questions 2.2.3 Demo videos 2.2.4 User manuals 2.2.5 Charts, Help windows, Videos 2.3 Prepare user training schedule 2.4 Practices when conducting user training	● Written tests
3. Gather user feedback	3.1 Method of gathering user feedback 3.1.1 Surveys 3.1.2 Form builders 3.1.3 Questionnaires 3.1.4 Observation 3.2 Preparation of Data collection tools 3.2.1 Surveys 3.2.2 Form builders 3.2.3 Questionnaires 3.2.4 Observation 3.3 Collection of User Feedback 3.3.1 Surveys 3.3.2 Feedback forms 3.3.3 Social media monitoring	● Practical Activities ● Project work ● Demonstration ● Group Work ● Role Plays ● Observation ● Third Party report ● Portfolio of Evidence ● Written tests

	3.3.4 Beta tests 3.3.5 User analytics 3.4 Customer feedback analysis 3.4.1 What it is 3.4.2 Why it's important 3.4.3 Feedback analysis methods	
4. Perform application maintenance	4.1 Carrying out technical assistance 4.1.1 Customer relation practices 4.2 Monitoring & reporting on application Performance 4.3 Performing application optimization 4.4 Performing security Application updates 4.4.1 Security update 4.4.2 System updates 4.5 Perform routine system maintenance 4.6 Performing system updates	● Practical Activities ● Project work ● Demonstration ● Group Work ● Role Plays ● Observation ● Third Party report ● Portfolio of Evidence ● Written tests

Suggested Delivery Methods

- Instructor led facilitation using active learning strategies
- Demonstration by trainer
- Practical work by trainees
- Role plays
- Project work by trainees
- Group Work
- Group projects

Recommended Resources for 25 Trainees

S/No.	Category/Item	Description/ Specifications	Quantity	Recommended Ratio (Item: Trainee)
A	**Learning Materials**			
1.	Internet connection	For each compute	1	1:1
2.	Flip charts	A1	1	1:25
3.	Textbooks	For reference	3	3:25
B	**Learning Facilities & infrastructure**			
4.	Computer Laboratory	To accommodate 25 Learners	1	1:25
5.	Theory Room	furnished with 25 Arm-chairs and a suitable trainer's table	1	1:25
C	**Consumable materials**			
6.	Printing papers	A4	2 Reams	1:12
7.	Toner / Ink bottles	For printers	2 pcs	1:12
8.	White board markers	Assorted colors	20	4:5
D	**Tools and Equipment**			
9.	Computers	Genuine Windows/Linux Genuine Microsoft office Software Google Workspace Account Antivirus Software Suitable IDE	25	1:1
10.	External storage media	HDD / SSD / Flash	1	1:25
11.	Printer	Working printer	2	1:12
12.	1 Smart-board / Smart TV / Projector (with screen)	Where available	1	1:25
13.	Whiteboard/Chalkboard	4 X 8 Feet	1	1:25

MODULE III

Unit Category	Unit Code	Unit Name	Unit Duration (Hours)
BASIC	0413 441 01A	ENTREPRENEURIAL SKILLS	40
CORE	0612 451 09A	COMPUTERIZED DATABASE SYSTEMS	340
		TOTAL	**380**

ENTREPRENEURIAL SKILLS

ISCED UNIT CODE: 0413 441 01A

TVET CDACC UNIT CODE: ICT/CU/SD/BC/01/5/MA

Relationship to occupational standards

This unit addresses the unit of competency: Apply Entrepreneurial skills.

Duration of unit: 40 hours

Unit Description:

This unit covers the competencies required to demonstrate an understanding of entrepreneurship. It involves demonstrating an understanding of financial literacy, applying entrepreneurial concepts identifying entrepreneurship opportunities, applying business legal aspects, and developing business innovative strategies and business plans.

Summary of Learning Outcomes

Learning Outcomes	Duration (Hours)
1. To apply financial literacy	6
2. To apply the entrepreneurial concept	4
3. To Identify entrepreneurship opportunities	6
4. To apply business legal aspects	6
5. To innovate Business Strategies	6
6. To develop business plan	12
TOTAL	**40**

Learning Outcomes, Content and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Apply financial literacy	1.1 Personal finance management 1.2 Balancing between needs and wants 1.3 Budget Preparation 1.4 Savings management 1.5 Factors to consider when deciding where to save 1.6 Debt management 1.7 Factors to consider before taking a loan 1.8 Investment decisions 1.9 Types of investments 1.10 Factors to consider when investing money 1.11 Insurance services ● Insurance products available in the market ● Insurable risks	● Observation ● Project ● Written assessment ● Oral assessment ● Third party report ● Interviews
2. Apply entrepreneurial concept	2.1 Difference between Entrepreneurs and Business persons 2.2 Types of entrepreneurs 2.3 Ways of becoming an entrepreneur 2.4 Characteristics of Entrepreneurs 2.5 salaried employment and self-employment 2.6 Requirements for entry into self-employment 2.7 Roles of an Entrepreneur in an enterprise 2.8 Contributions of Entrepreneurship	● Observation ● Project ● Written assessment ● Oral assessment ● Third party report

Learning Outcome	Content	Suggested Assessment Methods
3. Identify entrepreneurship opportunities	2.1 Sources of business ideas 2.2 Factors to consider when evaluating business opportunity 2.3 Business life cycle	● Observation ● Project ● Written assessment ● Oral assessment ● Third party report
4. Apply business legal aspects	4.1 Forms of business ownership 4.2 Business registration and licensing processing 4.3 Types of contracts and agreements 4.4 Employment laws 4.5 Taxation laws	● Observation ● Project ● Written assessment ● Oral assessment ● Third party report
5. Innovate business Strategies	5.1 Creativity in business 5.2 Innovative business strategies 5.3 Entrepreneurial Linkages 5.4 ICT in business growth and development	● Observation ● Project ● Written assessment ● Oral assessment ● Third party report
6. Develop Business Plan	6.1 Business description 6.2 Marketing plan 6.3 Organizational/Management plan 6.4 Production/operation plan 6.5 Financial plan 6.6 Executive summary 6.7 Business plan presentation 6.8 Business idea incubation	● Observation ● Written assessment ● Project ● Oral assessment ● Third party report

Suggested Methods of Instruction

- Direct instruction with active learning strategies
- Project (Business plan)
- Case studies
- Field trips

- Group Work
- Demonstration
- Question and answer
- Problem solving
- Experiential
- Team training
- Guest speakers

Recommended Resources for 25 trainees

S/No.	Category/Item	Description/ Specifications	Quantity	Recommended Ratio (Item: Trainee)
A	**Learning Materials**			
1.	Internet connection	For each computer	1	1:1
2.	Flip charts	A1	1	1:25
3.	Learning guide	For reference	3	3:25
B	**Learning Facilities & infrastructure**			
4.	Computer Laboratory	To accommodate 25 Learners	1	1:25
	Theory Room	furnished with 25 Arm-chairs and a suitable trainer's table	1	1:25
C	**Consumable materials**			
	Printing papers	A4	2 Reams	1:12
5.	Toner / Ink bottles	For printers	2 pcs	1:12
6.	White board markers	Assorted colors	20	4:5
D	**Tools and Equipment**			
	Computers	✓ Genuine Windows/Linux ✓ Genuine Microsoft office Software ✓ Google Workspace	25	1:1

		✓ Account Antivirus Software		
7.	External storage media	HDD / SSD / Flash	1	1:25
8.	Printer	Working printer	2	1:12
9.	1 Smart-board / Smart TV / Projector (with screen)	Where available	1	1:25

COMPUTERISED DATABASE SYSTEMS MANAGEMENT

ISCED UNIT CODE: 0612 451 09A

TVET CDACC UNIT CODE: ICT/CU/SD/CR/04/5/MA

Relationship to Occupational Standards

This unit addresses the Unit of Competency: **Manage Computerized Database System**

Duration of Unit: 340 Hours

UNIT DESCRIPTION

This unit covers the competencies required to manage computerized database system. It involves, designing database system, creating database system, Manipulating Computerized Database, managing database security and performing database maintenance

Summary of Learning Outcomes

Learning Outcomes	Duration (Hours)
1. To design database system	60
2. To create database system	100
3. To manipulate computerized database	90
4. To manage database security	50
5. To perform database maintenance	40
TOTAL	**340**

Learning Outcomes, Content and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Design database system	1.1. Identification of Database design approaches 1.1.1. Design approaches 1.1.1.1. Top – down design method 1.1.1.2. Bottom – up design method 1.1.1.3. Centralized design 1.1.1.4. Decentralized design 1.2. Identification of database design tools 1.2.1. Types of design tools 1.3. Database structures. 1.3.1. Database structure 1.3.2. Database models 1.3.2.1. Record-based model ● Hierarchical models ● Network Models ● Relational Models 1.3.2.2. Object-based data models ● Entity-Relationship (ER) ● Semantic ● Functional ● Object-oriented 1.3.2.3. Physical data models ● unifying model and ● the frame memory 1.3.3. Database schema 1.4. Database design architecture 1.4.1. Schema design 1.4.2. Database management system architecture 1.4.3. Data Warehousing and Big Data Architecture	❖ Practical Activities ❖ Project work ❖ Demonstration ❖ Group Work ❖ Observation ❖ Third Party report ❖ Portfolio of Evidence ❖ Written tests

	1.4.3.1. Multi-user DBMS architectures 1.4.3.2. Web service and service oriented Architectures 1.4.3.3. Distributed DBMS 1.4.4. Data Warehousing and Big Data Architecture 1.5. Database normalisation. 1.5.1. Types of normalisations 1.5.2. Process of normalization 1.6. Entity Relationship diagrams 1.7. Database design report 1.7.1. Key components of database design report	
2. Create database system	1.1. Database management software identification 1.1.1. Selecting Appropriate DBMS 1.2. Database development environment configuration. 1.2.1. Setting Up the Environment 1.2.2. Development Tools 1.3. Database objects 1.3.1. Tables 1.3.2. Indexes 1.3.3. Tools 1.3.4. Stored procedures and functions 1.4. Data attributes. 1.4.1. Types of attributes 1.5. Data relationships. 1.5.1. Types of relationships 1.5.1.1. One to one	❖ Practical Activities ❖ Project work ❖ Demonstration ❖ Group Work ❖ Observation ❖ Third Party report ❖ Portfolio of Evidence ❖ Written tests

	1.5.1.2. One to many 1.5.1.3. Many to many 1.6. Workplace safety and health practices 1.6.1. Importance 1.6.2. Digital safety 1.7. E-waste storage and disposal 1.8. E-waste management 1.8.1. Storage and Disposal 1.8.2. Erasure	
3. Manipulate computerized database	2.1. Database business rules 2.1.1. Unique constraints 2.1.2. Referential integrity 2.2. Inserting data into database 2.3. Insert statement in SQL 2.4. Data retrieval from the database 2.4.1. Selecting data from database 2.5. Data modification using queries. 2.5.1. Updating data in database 2.5.2. Modifying queries in database 2.6. Data deletion 2.6.1. Deleting data in table 2.6.2. Dropping database	❖ Practical Activities ❖ Project work ❖ Demonstration ❖ Group Work ❖ Third Party report ❖ Portfolio of Evidence ❖ Written tests
4. Manage database security	4.1 Database security risks identification 4.1.1 Common security risks 4.1.1.1 Unauthorized Access 4.1.1.2 SQL Injection Attacks 4.1.1.3 Data Leakage 4.1.1.4 Insider Threats 4.1.1.5 Weak Authentication and Access Control 4.1.1.6 Inadequate Patching	❖ Practical Activities ❖ Project work ❖ Demonstration ❖ Group Work ❖ Observation ❖ Third Party report ❖ Portfolio of Evidence ❖ Written tests

	4.1.1.7 Malware 4.2 Identification of database security control measures 4.2.1 Control measures 4.2.1.1 Access Control and Authentication 4.2.1.2 SQL Injection Prevention 4.2.1.3 Encryption 4.2.1.4 Regular Security Audits and Monitoring 4.2.1.5 Patching and Updates 4.2.1.6 Data Masking and Anonymization 4.2.1.7 Backup and recovery plans 4.2.1.8 User Activity Logging 4.2.1.9 Firewalls 4.3 Database security control measures implementation. 4.3.1 Policy Development 4.3.2 Access Management 4.3.3 Configuration Hardening 4.3.4 Testing 4.4 Carrying out Monitoring and auditing of database security 4.5 Performing database security documentation 4.5.1 Database maintenance schedule preparation 4.6 Training database users 4.6.1 End Users 4.6.2 Application Programmer or Specialized users or Back-End	

	Developer 4.6.3 System Analysts 4.6.4 Database Administrator (DBA) 4.6.5 Temporary Users or Casual Users	
5. Perform database maintenance	5.1 Database maintenance schedule preparation 5.1.1 Maintenance plans 5.1.2 Daily tasks 5.1.3 Weekly tasks 5.1.4 Monthly tasks 5.1.5 Quarterly tasks 5.1.6 Annual tasks 5.1.7 Database maintenance schedule preparation 5.1.7.1 Key elements in preparation of maintenance schedule 5.2 Database performance monitoring 5.2.1 Resource Utilization 5.2.2 Query Performance 5.2.3 Transaction Log Monitoring 5.2.4 Connection Monitoring 5.2.5 Alerting Systems 5.3 Database performance optimization 5.3.1 Query Optimization 5.3.2 Indexing Strategy 5.3.3 Partitioning 5.3.4 Configuration Tuning 5.3.5 Archiving 5.4 Database maintenance report generation 5.4.1 Report components 5.4.2 Report generation process	❖ Practical Activities ❖ Project work ❖ Demonstration ❖ Group Work ❖ Observation ❖ Third Party report ❖ Portfolio of Evidence ❖ Written tests

Suggested Delivery Methods

1. Instructor led facilitation using active learning strategies
2. Demonstration by trainer
3. Practical work by trainees
4. Viewing of related videos
5. Field Visits
6. Group Work
7. Role plays
8. Group projects

Recommended Resources for 25 Trainees

S/No.	Category/Item	Description/ Specifications	Quantity	Recommended Ratio (Item: Trainee)
A	**Learning Materials**			
	Internet connection	✓ For each compute	1	1:1
2.	Flip charts	A1	1	1:25
3.	Textbooks	For reference	3	3:25
B	**Learning Facilities & infrastructure**			
4.	Computer Laboratory	To accommodate 25 Learners	1	1:25
5.	Theory Room	furnished with 25 Arm-chairs and a suitable trainer's table	1	1:25
C	**Consumable materials**			
6.	Printing papers	A4	2 Reams	1:12
7.	Toner / Ink bottles	For printers	2 pcs	1:12
8.	White board markers	Assorted colors	20	4:5
D	**Tools and Equipment**			
9.	Computers	✓ Genuine Windows/Linux ✓ Genuine Microsof office Software ✓ Google Workspace	25	1:1

		☐ Account ✓ Antivirus Software ✓ Suitable IDE		
10.	External storage media	HDD / SSD / Flash	1	1:25
11.	Printer	Working printer	2	1:12
12.	1 Smart-board / Smart TV / Projector (with screen)	Where available	1	1:25
13.	Whiteboard/Chalkboard	4 X 8 Feet	1	1:25

MODULE IV

Unit Category	Unit Code	Unit Name	Unit Duration (Hours)
COMMON	0688 451 05A	PROJECT MANAGEMENT PRINCIPLES	110
BASIC	0417 441 02 A	WORK ETHICS AND PRACTICES	40
CORE	0613 451 10A	WEB APPLICATION DEVELOPMENT	250
		TOTAL	**400**

PROJECT MANAGEMENT PRINCIPLES

ISCED UNIT CODE: 0688 451 05A

TVET CDACC UNIT CODE: ICT/CU/SD/CC/03/5/MA

Relationship to Occupational Standards

This unit addresses the Unit of Competency: **Apply Project Management Principles**

Duration of Unit: 110 Hours

Unit Description

This unit covers the competencies required to apply project management principles. It involves executing project initiation, performing project planning, performing project monitoring, and performing project closure.

Summary of Learning Outcomes

Learning Outcomes	Duration (Hours)
1. To execute project initiation	20
2. To perform project planning	40
3. To perform project monitoring	40
4. To perform project closure	10
TOTAL	**110**

Learning Outcomes, Content and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Execute project initiation	1.1. Identification of Project scope 1.1.1. Key Components Listing 1.1.1.1.Project goals and objectives 1.1.1.2. Deliverables 1.1.1.3.Inclusions and exclusions 1.1.1.4. Constraints and assumptions 1.1.2. Scope formulation process 1.1.2.1. Review client or stakeholder needs. 1.1.2.2. Document the project requirements. 1.1.2.3. Define project boundaries 1.1.2.4. Confirm scope with stakeholders 1.2. Determination of Project deliverables 1.2.1. Internal Deliverables 1.2.2. External Deliverables 1.2.3. Steps to Determine project deliverables: 1.2.3.1. Identifying key outcomes 1.2.3.2. Breaking down the project 1.2.3.3. Defining milestones	● Practical Activities ● Project work ● Demonstration ● Group Work ● Observation ● Third Party report ● Portfolio of Evidence ● Written tests

	1.2.3.4. Clarifying project acceptance criteria 1.3. Identification of project objectives 1.3.1. SMART Objectives: 1.4. Project initiation document (PID) preparation. 1.4.1. Key Sections of the PID: 1.4.1.1. Project Purpose and Justification 1.4.1.2. Project Scope 1.4.1.3. Project Deliverables 1.4.1.4. Project Objectives: 1.4.1.5. Stakeholder Analysis 1.4.1.6. Project Organization 1.4.1.7. Project Timeline and Milestones 1.4.1.8. Budget and Resource Plan 1.4.1.9. Risk Assessment 1.4.1.10. Quality Management 1.4.2. Importance of the PID:	
2. Perform project planning	2.1. Preparation of Project budget 2.1.1. Steps of preparing budget 2.1.2. Budgeting techniques 2.2. Determination of Project schedule 2.2.1. Steps of preparing schedule 2.2.2. Project scheduling tools 2.2.2.1. Critical path method 2.2.2.2. Program Evaluation	● Practical Activities ● Project work ● Demonstration ● Group Work ● Observation ● Third Party report ● Portfolio of Evidence ● Written tests

	and Review Technique (PERT) 2.3. Allocation of Project resources 2.3.1. Identify resources 2.3.2. Allocate based on availability 2.3.3. Balance resources 2.3.4. Monitor and adjust 2.4. Determination Project work breakdown structures (WBS) 2.4.1. Identify major project deliverables 2.4.2. Divide deliverables into smaller tasks 2.4.3. Assign resources and timelines 2.4.4. Use a numbering system 2.5. Preparation of Project quality plan 2.5.1. Quality objectives 2.5.2. Quality assurance activities 2.5.3. Quality control measures 2.5.4. Acceptance criteria 2.6. Formation of Project team. 2.6.1. Identify required skills 2.6.2. Select team members 2.6.3. Define roles and responsibilities 2.6.4. Build team collaboration 2.7. Project team Roles and responsibilities 2.7.1. Define responsibilities	

	2.7.2. Assign roles 2.7.3. Document roles and responsibilities 2.7.4. Get sign-off 2.8. Preparation of Project plan. 2.8.1. Project objectives 2.8.2. Scope statement 2.8.3. Schedule 2.8.4. Resource plan 2.8.5. Risk management plan 2.8.6. Communication plan 2.8.7. Change management plan 2.8.8. Project Approval	
3. Perform project monitoring	3.1. Tracking Project costs 3.1.1. Steps to Track Costs: 3.1.1.1. Set a baseline budget: 3.1.1.2. Record actual expenses: 3.1.1.3. Compare planned vs. actual costs 3.1.1.4. Use Earned Value Management 3.1.1.5. Report cost status 3.1.2. Tools for Tracking Costs 3.2. Monitoring Project deliverables and objectives 3.2.1. Define key deliverables 3.2.2. Set up a monitoring schedule 3.2.3. Track against the project plan	● Practical Activities ● Project work ● Demonstration ● Group Work ● Observation ● Third Party report ● Portfolio of Evidence ● Written tests

	3.2.4. Use performance indicators 3.2.5. Report to stakeholders 3.2.6. ISO quality Standards 3.3. Monitoring Project team performance 3.3.1. Steps to Monitor Team Performance 3.4. Assessing Project risks 3.4.1. Steps to Assess Risks: 3.4.1.1. Identify potential risks 3.4.1.2. Evaluate risk impact and probability 3.4.1.3. Prioritize risks 3.4.1.4. Conduct a SWOT analysis 3.4.2. Types of Project Risks: 3.4.2.1. Technical risks 3.4.2.2. Schedule risks 3.4.2.3. Budget risks 3.4.2.4. Resource risks. 3.5. Managing Project risks. 3.5.1.1. Steps to manage risks	
4. Perform project closure	4.1. Performing Project review 4.1.1. Key Objectives of a Project Review 4.1.2. Steps to Perform a Project Review 4.1.3. Post-Project Review Techniques 4.2. Review of Final project budget 4.2.1. Steps to Review the Final	• Practical Activities • Project work • Demonstration • Group Work • Observation • Third Party report • Portfolio of Evidence • Written tests

	Budget 4.2.2. Key Considerations 4.3. Preparation of detailed project review report 4.3.1. Key Components of the Project Review Report 4.3.2. Purpose of the report	

Suggested Delivery Methods

- Instructor led facilitation using active learning strategies
- Demonstration by trainer
- Practical work by trainees
- Viewing of related videos
- Field Visits
- Group Work
- Role plays
- Group projects

S/No.	Category/Item	Description/ Specifications	Quantity	Recommended Ratio (Item: Trainee)
A	**Learning Materials**			
	Internet connection	✓ For each computer	1	1:1
	Flip charts	A1	1	1:25
	Textbooks	For reference	3	3:25
B	**Learning Facilities & infrastructure**			
	Theory Room	furnished with 25 Arm-chairs and a suitable trainer's table	1	1:25
C	**Consumable materials**			
	Printing papers	A4	2 Reams	1:12
	White board markers	Assorted colors	20	4:5
D	**Tools and Equipment**			

	Computers	✓ Genuine Windows/Linux ✓ Genuine Microsoft office Software ✓ Google Workspace Account ✓ Antivirus Software ✓ MS Project / Ms Visio ✓ Visual Paradigm ✓ Clickup	25	1:1
	External storage media	HDD / SSD / Flash	1	1:25
	Printer	Working printer	2	1:12
	1 Smart-board / Smart TV / Projector (with screen)	Where available	1	1:25
	Whiteboard/Chalkboard	4 X 8 Feet	1	1:25

WORK ETHICS AND PRACTICES

ISCED UNIT CODE: 0417 441 02 A

TVET CDACC UNIT CODE: ICT/CU/SD/BC/02/5/MA

Relationship to Occupational Standards

This unit addresses the Unit of Competency: Apply work ethics and practices.

Duration of Unit: 40 hours

Unit Description

This unit covers competencies required to apply work ethics and practices. It involves the ability to: conduct self-management, promote ethical work practices and values, promote teamwork, manage workplace conflicts, maintain professional and personal development, apply problem-solving, and promote customer care.

Summary of Learning Outcomes

Learning Outcomes	Duration (Hours)
1. To apply self-management skills	10
2. To promote ethical practices and values	4
3. To promote teamwork	10
4. To maintain professional and personal development	10
5. To apply problem-solving skills	4
6. To promote customer care.	2
TOTAL	**40**

Learning Outcomes, Content, and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Apply self-management skills	1.1 Self-awareness 1.2 Formulating personal vision, mission, and goals 1.3 Healthy lifestyle practices 1.4 Strategies for overcoming work challenge 1.5 Emotional intelligence 1.6 Coping with Work Stress. 1.7 Assertiveness versus aggressiveness and passiveness 1.8 Developing and maintaining high self-esteem 1.9 Developing and maintaining positive self-image 1.10 Time management 1.11 Setting performance targets 1.12 Monitoring and evaluating performance targets	● Observation ● Written assessment ● Oral assessment ● Third party reports ● Portfolio of evidence ● Project ● Practical
2. Promote ethical work practices and values	2.1 Integrity 2.2 Core Values, ethics and beliefs 2.3 Patriotism 2.4 Professionalism 2.5 Organizational codes of conduct 2.6 Industry policies and procedures	● Observation ● Written assessment ● Oral assessment ● Third party reports ● Portfolio of evidence ● Project ● Practical
3. Promote Teamwork	3.1 Types of teams 3.2 Team building 3.3 Individual responsibilities in a	● Observation ● Written assessment

Learning Outcome	Content	Suggested Assessment Methods
	team 3.4 Determination of team roles and objectives 3.5 Team parameters and relationships 3.6 Benefits of teamwork 3.7 Qualities of a team player 3.8 Leading a team 3.9 Team performance and evaluation 3.10 Conflicts and conflict resolution 3.11 Gender and diversity mainstreaming 3.12 Developing Healthy workplace relationships 3.13 Adaptability and flexibility 3.14 Coaching and mentoring skills	● Oral assessment ● Third party reports ● Portfolio of evidence ● Project ● Practical
4. Maintain professional and personal development	4.1 Personal vs professional development and growth 4.2 Avenues for professional growth 4.3 Recognizing career advancement 4.4 Training and career opportunities 4.5 Assessing training needs 4.6 Mobilizing training resources 4.7 Licenses and certifications for professional growth and development 4.8 Pursuing personal and organizational goals 4.9 Managing work priorities and	● Observation ● Written assessment ● Oral assessment ● Third party reports ● Portfolio of evidence ● Project ● Practical

Learning Outcome	Content	Suggested Assessment Methods
	commitments 4.10 Dynamism and on-the-job learning	
5. Apply Problem-solving skills	5.1 Causes of problems 5.2 Methods of solving problems 5.3 Problem-solving process 5.4 Decision making 5.5 Creative thinking and critical thinking process in development of innovative and practical solutions	● Observation ● Written assessment ● Oral assessment ● Third party reports ● Portfolio of evidence ● Project ● Practical
6. Promote Customer Care	6.1 Identifying customer needs 6.2 Qualities of good customer service 6.3 Customer feedback methods 6.4 Resolving customer concerns 6.5 Customer outreach programs 6.6 Customer retention	● Observation ● Written assessment ● Oral assessment ● Third party reports ● Portfolio of evidence ● Project ● Practical

Suggested Methods of Instruction

- Instructor lead facilitation of theory using active learning strategies.
- Demonstrations
- Simulation/Role play
- Group Discussion
- Presentations
- Projects
- Case studies
- Assignments

Recommended Resources for 25 Trainees

S/No.	Category/Item	Description/ Specifications	Quantity	Recommended Ratio (Trainee: Item)
A	**Learning Materials**			
1.	Textbooks		5 pcs	5:1
2.	PowerPoint presentations	For trainer's use		
3.	Assorted colour of whiteboard markers	For trainer's use	2 packets	
4.	e-Didactics	For trainer's use		
5.	Flashcards			
6.	Flip charts			
7.	Whiteboard			
B	**Learning Facilities & infrastructure**			
8.	Lecture/theory room		1	25:1
C	**Consumable materials**			
9.	Printing Papers		1 ream	1:20
10.	Toners		2 pcs	13:1
11.	Internet connection			
D	**Tools and Equipment**			
12.	Projectors		1	25:1
13.	Printers		4	6:1
14.	Computers/Mobile Phones		25 pcs	1:1

WEB APPLICATION DEVELOPMENT

ISCED UNIT CODE: 0613 451 10A

TVET CDACC UNIT CODE: ICT/CU/SD/CR/05/5/MA

Relationship to Occupational Standards

This unit addresses the Unit of Competency: Create Web Application

Duration of Unit: 250 Hours

UNIT DESCRIPTION

This unit covers the competencies required to create web application. It involves, designing web application, writing web application source code, testing web application, debugging web application and hosting web application

Summary of Learning Outcomes

Learning Outcomes	Duration (Hours)
1. To design web application	40
2. To write web application source code	100
3. To test web application	30
4. To debug web application	40
5. To host web application	40
TOTAL	**250**

Learning Outcomes, Content and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Design web application	1.1. Design Principles 1.1.1. User-centered Design 1.1.1.1. User's goals 1.1.1.2. User's tasks 1.1.1.3. Preferences 1.1.2. Information Architecture 1.1.3. Visual Design 1.1.3.1. Layout, 1.1.3.2. Typography, 1.1.3.3. Colour scheme, 1.1.3.4. Imagery. 1.1.4. Interaction Design 1.1.5. Accessibility 1.1.6. Web application design tools 1.1.6.1. Drafting application e.g Figma, illustrator or Photoshop basics 1.2. Designing of web application functionality 1.2.1. Creation of website application site map 1.2.2. Performance 1.3. Creation of web application interface design 1.3.1. Responsiveness of web pages 1.4. Designing of web application output 1.4.1. Rendering pdf, Excel documents	• Practical Activities • Project work • Demonstration • Group Work • Observation • Third Party report • Portfolio of Evidence • Written tests

2. Write web application source code	2.1. Creation of user interface development tools. 2.1.1. Identification of front-End development tools 2.1.1.1 Hypertext Markup Language 2.1.1.1. HTML Tags, elements and attributes 2.1.1.2 Cascading Style Sheets 2.1.1.2. Embedded CSS 2.1.1.3. Inline CSS 2.1.1.4. External 2.1.1.3 JavaScript basics 2.1.1.5. Data types operation 2.1.1.6. Document Object Model (DOM) 2.1.1.7. JavaScript Frameworks overview e.g JQuery syntax and events 2.2. Performing Version control 2.3. Functionality and interactivity development 2.4. Responsive design implementation Methodology 2.4.1. Screen resolutions 2.5. Front-end Development 2.5.1. Hypertext Markup Language 2.5.2. Cascading Style Sheets 2.5.3. JavaScript basics 2.5.4. JavaScript Frameworks overview 2.6. Application Programming Interfaces (API) 2.6.1. API integration	• Practical Activities • Project work • Demonstration • Group Work • Observation • Third Party report • Portfolio of Evidence • Written tests

	2.6.2. Application API testing 2.7. Server-side coding tools 2.7.1. PHP coding basics 2.8. Back-end Development 2.8.1. Server-side coding tools e.g PHP coding basics 2.8.2. Back-end frameworks 2.9. Back-end database implementation 2.9.1. Database creation 2.9.2. Database connection 2.9.3. Database Manipulation through a web interface 2.10. Back-end API creation 2.11. Workplace safety and health practices 2.11.1. Applicable OSHA regulations and laws 2.12. E-waste storage and disposal 2.13. E-waste management	
3. Test web application	3.1. Web application testing Types 3.2. Web application test plan 3.2.1. Test environment 3.2.2. Test scope 3.2.3. Schedule 3.3. Web application testing tools 3.3.1. Performance testing tools 3.3.2. functional testing tools 3.3.3. security testing tools 3.3.4. Cross-browser testing 3.3.5. Mobile-web application testing tools 3.3.6. Usability testing tools e,g Google analytics	• Practical Activities • Project work • Demonstration • Group Work • Third Party report • Portfolio of Evidence • Written tests

	3.4. Test report preparation 3.4.1. Types of Test reports	
4. Debug web application	4.1. revise source code for errors i. Types of Errors 1. Logical 2. Runtime 3. Syntax b. Apply debugging tools; Error handling techniques i. Source code revision ii. Debugging tools c. Performing regression testing	• Practical Activities • Project work • Demonstration • Group Work • Observation • Third Party report • Portfolio of Evidence • Written tests
5. Host web application	a. Web hosting service identification i. Types of web hosting techniques e.g Shared, Dedicated, Free, Cloud. b. Domain acquisition and configuration i. Domain lookup ii. Domain Registration iii. Domain Pricing c. Web Server configuration i. Domain Configuration ii. DNS iii. Control panel d. Web application deployment tools i. FTP ii. GITs iii. Docker e. Web security measures	• Practical Activities • Project work • Demonstration • Group Work • Observation • Third Party report • Portfolio of Evidence • Written tests

	i. SSL / TLS Certificates ii. Firewalls iii. Updates and upgrades iv. Strong authentication v. Backup solutions f. Web application maintenance and monitoring i. Server Maintenance ii. Database Maintenance iii. Application Updates iv. Monitoring tools 1. Performance monitoring 2. Up-time monitoring 3. Security monitoring 4. Log monitoring	

Suggested Delivery Methods

- Instructor led facilitation using active learning strategies
- Demonstration by trainer
- Practical work by trainees
- Viewing of related videos
- Field Visits
- Group Work
- Group projects

Recommended Resources for 25 Trainees

S/No.	Category/Item	Description/ Specifications	Quantity	Recommended Ratio (Item: Trainee)
A	**Learning Materials**			
1.	Internet connection	✓ For each computer	1	1:1
2.	Flip charts	A1	1	1:25
3.	Textbooks	For reference	3	3:25
B	**Learning Facilities & infrastructure**			
4.	Computer Laboratory	To accommodate 25 Learners	1	1:25
5.	Theory Room	furnished with 25 Arm-chairs and a suitable trainer's table	1	1:25
C	**Consumable materials**			
6.	Printing papers	A4	2 Reams	1:12
7.	Toner / Ink bottles	For printers	2 pcs	1:12
8.	White board markers	Assorted colors	20	4:5
D	**Tools and Equipment**			
9.	Computers	✓ Genuine Windows/Linux ✓ Genuine Microsoft office Software ✓ Google Workspace Account ✓ Antivirus Software ✓ Suitable IDE	25	1:1
10.	External storage media	HDD / SSD / Flash	1	1:25
11.	Printer	Working printer	2	1:12
12.	1 Smart-board / Smart TV /	Where available	1	1:25

	Projector (with screen)			
13.	Whiteboard/Chalkboard	4 X 8 Feet	1	1:25

MODULE V

Unit Category	Unit Code	Unit Name	Unit Duration (Hours)
CORE	0613 551 02 A	OBJECT ORIENTED PROGRAMMING	170
CORE	0613 551 03 A	DESKTOP APPLICATION DEVELOPMENT	240
		Total	**410**

OBJECT ORIENTED PROGRAMMING

ISCED UNIT CODE: 0613 551 02 A

TVET CDACC UNIT CODE: ICT/CU/SD/CR/01/6/MA

Relationship to Occupational Standards

This unit addresses the Unit of Competency: Apply Object Oriented Programing

Duration of Unit: 170 Hours

Unit Description

This unit covers the competencies required to apply computer programming principles. It involves applying computer programming skills, demonstrating structured programming skills and demonstrating object oriented programming skills.

Summary of Learning Outcomes

Learning Outcomes	Duration (Hours)
1. To apply computer programming skills	**30**
2. To demonstrate structured programming skills	**40**
3. To demonstrate object-oriented programming skills	**100**
TOTAL	**170**

Learning Outcomes, Content and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Apply Computer Programming Skills	1.1 Programming Language Types 1.1.1 Overview of programming language categories 1.1.1.1. procedural	● Practical Activities ● Project work ● Group work ● Observation

	1.1.1.2. object-oriented 1.1.1.3. functional 1.1.2 Criteria for selecting languages based on user requirements 1.2 Programming Paradigms 1.2.1 Explanation of common programming paradigms (e.g., imperative, declarative) 1.2.2 Choosing the appropriate paradigm based on project needs 1.3 Program Development Life Cycle 1.3.1 Stages of the program development life cycle (e.g., planning, design, implementation) 1.3.2 Best practices for adapting the life cycle to work requirements 1.4 Program Design Tools 1.4.1 Overview of design tools (e.g. algorithms, flowcharts, wireframes, pseudocodes, decision table/trees) 1.4.2 Selecting design tools based on user requirements and project complexity 1.5 Program Writing Tools 1.5.1 Common program writing tools and IDEs (e.g.Visual	• Portfolio of Evidence • Written tests

	Studio, codeblocks, DEV C++, Eclipse) 1.5.2 Evaluating tools based on system requirements and developer preferences	
2. Demonstrate structured programming skills	2.1 Declaring Identifiers 2.1.1 Guidelines for naming conventions and best practices 2.1.2 Ensuring identifiers align with program design specifications 2.1.3 Creating a naming convention guide for different types of identifiers. 2.1.4 Evaluating identifier 2.2 Initializing Variables and Constants 2.2.1 Importance of proper initialization in programming 2.2.2 Techniques for initialization based on design specifications 2.2.3 Writing code snippets demonstrating correct and incorrect initialization. 2.2.4 Conducting a workshop on variable and constant initialization techniques. 2.3 Applying Data Control	• Practical Activities • Project work • Group work • Observation • Portfolio of Evidence • Written tests

	Structures 2.3.1 Types of control structures (e.g., loops, conditionals) 2.3.2 Best practices for implementing control structures as per design requirements 2.3.3 Solving coding challenges that require the use of different control structures. 2.3.4 Creating flowcharts to visually represent control structures in a program. 2.4 Applying Data Structures 2.4.1 Overview of common data structures (e.g., arrays, linked lists) 2.4.2 Selecting appropriate data structures based on design specifications 2.4.3 Implementing various data structures in a programming language of choice. 2.4.4 Comparing performance metrics of different data structures in a small project. 2.5 Creating Subroutines 2.5.1 Benefits of using subroutines (e.g., modularity, reusability)	

	2.5.2 Designing subroutines to meet user needs 2.5.3 Functions and subprograms 2.5.4 Design and implement a subroutine library for common tasks. 2.5.5 Program to create subroutines based on given specifications. 2.6 Applying User-Defined Data Types 2.6.1 Overview of user-defined data types (e.g., structs, classes) 2.6.2 Criteria for selecting data types based on system requirements 2.6.3 Designing a class or struct for a real-world object 2.6.4 Collaborating on a group project that requires the use of user-defined data types. 2.7 Performing Debugging 2.7.1 Common debugging techniques and tools 2.7.2 Participating in a debugging workshop using a sample project. 2.7.3 Creating a debugging checklist based on work procedures. 2.8 Compiling Program	

	2.8.1 Steps involved in the compilation process 2.8.2 Ensuring compliance with system requirements during compilation 2.8.3 Compiling a series of programs with intentional errors to learn about error messages. 2.8.4 Researching and presenting on different compilers and their features.	
3. Demonstrate object-oriented programming skills	1.1 Implementing Objects and Classes 1.1.1 Overview of objects and classes in OOP 1.1.2 Implementation of classes 1.1.3 Creating a simple class and instantiating objects to demonstrate understanding. 1.1.4 Participating in a group project where each member implements a different class based on a shared design. 1.2 Declaring Object Methods 3.2.1 Defining class methods that fulfill application requirements 3.2.2 Best practices for method naming and functionality 3.2.3 Developing a class with various methods and demonstrating their usage	● Practical Activities ● Project work ● Group work ● Observation ● Portfolio of Evidence ● Written tests

	in a small application. 3.2.4 Conducting a code review session focusing on method declarations and their alignment with application requirements. 3.3 Applying Namespaces 3.3.1 Understanding the role of namespaces in OOP 3.3.2 Implementing namespaces as per work procedures 3.3.3 Create a project that uses multiple namespaces to organize code effectively. 3.3.4 Explore and present on the benefits of using namespaces in a collaborative coding environment. 3.4 Applying Data Abstraction Concepts 3.4.1 Definition and importance of data abstraction 3.4.2 Implementing abstraction in line with work procedures 3.4.3 Designing an abstract class and demonstrate its use in a program. 3.4.4 Trainees work in pairs to identify and implement abstraction in existing codebases. 3.5 Applying Object Encapsulation 3.5.1 Understanding encapsulation and its	

	significance 3.5.2 Implementing encapsulation in programs 3.5.3 Type of access modifiers 3.5.3.1 Private 3.5.3.2 Public 3.5.3.3 Protected 3.5.4 A class to demonstrate encapsulation by using private and public access modifiers. 3.5.5 Class presentation on advantages of encapsulation in a program. 3.6 Implementing Class Templates 3.6.1 Overview of class templates and their applications 3.6.2 Creating class templates 3.6.3 Write a generic class template and demonstrate its usage with different data types. 3.6.4 Collaborating on a project that requires the use of class templates for various functionalities. 3.7 Implementing Class Inheritance 3.7.1 Inheritance concepts and types of inheritance (single, multiple) 3.7.2 Applying inheritance in programs 3.7.3 Creating a class hierarchy to	

	demonstrate inheritance concepts. 3.7.4 Participating in a coding challenge that requires implementing inheritance in a given scenario. 3.8 Implementing Polymorphism 3.8.1 Definition and types of polymorphism (compile-time vs. runtime) 3.8.2 Implementing polymorphism in a program 3.8.3 Develop a program to demonstrate both types of polymorphism. 3.8.4 Engage in a discussion or workshop on the practical applications and benefits of polymorphism in software development.	

Suggested Delivery Methods

- Instructor led facilitation using active learning strategies
- Demonstration by trainer
- Practical work by trainees
- Viewing of related videos
- Field Visits
- Group Work
- Role plays
- Group projects

Recommended Resources for 25 Trainees

S/No.	Category/Item	Description/	Quantity	Recommend

		Specifications		ed Ratio (Item: Trainee)
A	**Learning Materials**			
14.	Internet connection	✓ For each computer	1	1:1
15.	Flip charts	A1	1	1:25
16.	Textbooks	For reference	3	3:25
B	**Learning Facilities & infrastructure**			
17.	Computer Laboratory	To accommodate 25 Learners	1	1:25
18.	Theory Room	furnished with 25 Arm-chairs and a suitable trainer's table	1	1:25
C	**Consumable materials**			
19.	Printing papers	A4	2 Reams	1:12
20.	Toner / Ink bottles	For printers	2 pcs	1:12
21.	White board markers	Assorted colors	20	4:5
D	**Tools and Equipment**			
22.	Computers	✓ Genuine Windows/Linux ✓ Genuine Microsoft office Software ✓ Google Workspace Account ✓ Antivirus Software ✓ Suitable IDE	25	1:1
23.	External storage media	HDD / SSD / Flash	1	1:25
24.	Printer	Working printer	2	1:12
25.	1 Smart-board / Smart TV / Projector (with screen)	Where available	1	1:25
26.	Whiteboard/Chalkboard	4 X 8 Feet	1	1:25

DESKTOP APPLICATION DEVELOPMENT

ISCED UNIT CODE: 0613 551 03 A

TVET CDACC UNIT CODE: ICT/CU/SD/CR/02/6/MA

Relationship to Occupational Standards

This unit addresses the Unit of Competency: Develop Desktop Application

Duration of Unit: 240 Hours

UNIT DESCRIPTION

This unit covers the competencies required to develop desktop application. It involves designing desktop application, writing desktop application source code, debugging desktop application, testing desktop application and deploying desktop application.

Summary of Learning Outcomes

Learning Outcomes	Duration (Hours)
1. To design desktop application	60
2. To write desktop application source code	60
3. To debug desktop application	40
4. To test desktop application	50
5. To deploy desktop application	30
TOTAL	**240**

Learning Outcomes, Content and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Design desktop application	1.1. Desktop application design tools 1.1.1. Basic desktop application programming concepts 1.1.1.1. Events 1.1.1.2. Objects 1.1.1.3. Controls 1.1.1.4. Methods 1.1.2. Application development stages 1.1.3. Identifying Desktop Application Design Tools 1.1.4. Overview of desktop application design tools 1.1.5. Criteria for selecting tools 1.1.6. Case studies of tool selection in real-world applications 1.2. Designing Application Functionality 1.2.1. Techniques for gathering user requirements 1.2.2. Mapping user needs to application features 1.2.3. Prioritizing functionality based on user feedback 1.3. Creating the Application Interface 1.3.1. Principles of effective user interface design 1.3.2. Creating wireframes and prototypes 1.3.3. User testing and feedback incorporation	● Practical Activities ● Project work ● Demonstration ● Group Work ● Observation ● Portfolio of Evidence ● Written tests

	1.4. Designing Application Output 1.4.1. Types of application output (reports, visualizations) 1.4.2. Qualities of a good system user output 1.4.3. Methods for testing and validating system output	
2. Write desktop application source code	2.1. Identifying Development Tools 2.1.1. Overview of desktop application development tools 2.1.2. Visual Studio, Netbeans, JetBrains 2.1.3. Criteria for selecting tools based on system requirements 2.1.4. Examples of popular development environments (e.g., Visual Studio, Netbeans, JetBrains) 2.1.5. Parts of Integrated Development Environment 2.1.6. Types of controls and objects 2.1.6.1. Buttons 2.1.6.2. Textboxes 2.1.6.3. Labels 2.1.6.4. Combobox 2.1.6.5. Datagrid 2.1.6.6. Listview 2.1.6.7. Forms 2.2. Developing Application Interface 2.2.1. Implementing the interface design as per specifications	● Practical Activities ● Project work ● Demonstration ● Group Work ● Observation ● Portfolio of Evidence ● Written tests

	2.2.2. Using design patterns and frameworks (e.g., MVC) 2.2.3. Ensuring user experience (UX) best practices are followed 2.3. Designing the Database 2.3.1. Understanding user needs for data management 2.3.2. Choosing the right database model (e.g., relational vs. NoSQL) 2.3.3. Creating data schemas and relationships 2.4. Database Integration 2.4.1. Techniques for integrating databases with applications 2.4.2. Ensuring data consistency and integrity eg datafield constraints 2.4.3. Using APIs and ORM (Object-Relational Mapping) tools 2.5. Implementing Application Functionality 2.5.1. Writing source code to meet user requirements 2.5.2. Testing functionality through unit and integration tests 2.5.3. Iterative development and user feedback incorporation 2.6. Observing Workplace Safety and Health Practices 2.6.1. Overview of OSHA regulations relevant to software development 2.6.2. Best practices for maintaining a safe workplace 2.6.3. Importance of ergonomics and safe equipment usage	

	2.7. Identifying E-Waste Storage and Disposal Methods 2.7.1. Understanding e-waste regulations as per OSHA 2.7.2. Proper storage techniques for electronic waste 2.7.3. Safe disposal methods and recycling options 2.8. Demonstrating E-Waste Management 2.8.1. Implementing e-waste management practices in the workplace 2.8.2. Training staff on e-waste handling and disposal 2.8.3. Monitoring and reporting e-waste management efforts	
3. Debug desktop application	3.1. Checking Source Code for Bugs and Errors 3.1.1. Techniques for static code analysis 3.1.2. Manual code review practices 3.2. Performing Debugging with Tools 3.2.1. Overview of popular debugging tools (e.g., GDB, Visual Studio Debugger) 3.2.2. Best practices for using debugging tools effectively 3.2.2.1. Breakpoints 3.2.2.2. Step options 3.2.2.3. Running commands eg Break, Pause & Stop 3.2.2.4. Examining variables & expressions 3.3. Conducting Regression Testing 3.3.1. Types of regression testing (e.g., automated vs. manual) 3.3.2. Strategies for identifying test cases	• Practical Activities • Project work • Demonstration • Group Work • Observation • Portfolio of Evidence • Written tests

	for regression 3.4. Documenting Source Code Changes 3.4.1. Importance of version control systems (e.g., Git) 3.4.2. Best practices for maintaining clear and concise documentation	
4. Test desktop application	4.1. Identifying Testing Types 4.1.1. Testing types 4.1.1.1. Unit test 4.1.1.2. Integration test 4.1.1.3. Usability test 4.1.1.4. System testing 4.1.1.5. Security test 4.1.1.6. Performance test 4.1.1.7. Compatibility test 4.1.2. Selecting appropriate testing types based on user requirements 4.2. Preparing Test Plan 4.2.1. Key components of a test plan (e.g., objectives, scope, resources) 4.2.2. Best practices for aligning the test plan with work procedures 4.3. Executing Tests as per Test Plan 4.3.1. Methods for executing various tests (manual vs. automated) 4.3.2. Tracking test progress and issues during execution 4.4. Preparing the Test Report 4.4.1. Essential elements of a test report (e.g., findings, recommendations) 4.4.2. Best practices for documenting results according to work procedures	• Practical Activities • Project work • Demonstration • Group Work • Observation • Portfolio of Evidence • Written tests
5. Deploy desktop	5.1. Identifying Deployment Strategy 5.1.1. Overview of deployment strategies	• Practical Activities

applicatio n	(e.g., phased, big bang) 5.1.2. Factors influencing the choice of deployment strategy based on work procedures 5.2. Identifying Deployment Tools 5.2.1. Overview of popular deployment tools (e.g., InstallShield, NSIS) 5.2.2. Evaluating tools based on user requirements and application needs 5.3. Packaging the Application 5.3.1. Best practices for creating installation packages 5.3.2. Ensuring compliance with application requirements (e.g., dependencies, configurations) 5.4. Distributing the Application 5.4.1. Methods for distributing applications (e.g., direct download, app stores) 5.4.2. Ensuring distribution aligns with established work procedures	• Project work • Demonstration • Group Work • Observation • Portfolio of Evidence • Written tests

Suggested Delivery Methods

- Instructor led facilitation using active learning strategies
- Demonstration by trainer
- Practical work by trainees
- Viewing of related videos
- Field Visits
- Group Work
- Role plays
- Group projects

Recommended Resources for 25 Trainees

S/No.	Category/Item	Description/ Specifications	Quantity	Recommended Ratio (Item: Trainee)
A	**Learning Materials**			
27.	Internet connection	✓ For each computer	1	1:1
28.	Flip charts	A1	1	1:25
29.	Textbooks	For reference	3	3:25
B	**Learning Facilities & infrastructure**			
30.	Computer Laboratory	To accommodate 25 Learners	1	1:25
31.	Theory Room	furnished with 25 Arm-chairs and a suitable trainer's table	1	1:25
C	**Consumable materials**			
32.	Printing papers	A4	2 Reams	1:12
33.	Toner / Ink bottles	For printers	2 pcs	1:12
34.	White board markers	Assorted colors	20	4:5
D	**Tools and Equipment**			
35.	Computers	✓ Genuine Windows/Linux ✓ Genuine Microsof office Software ✓ Google Workspac Account ✓ Antivirus Softwar ✓ Suitable IDE	25	1:1
36.	External storage media	HDD / SSD / Flash	1	1:25
37.	Printer	Working printer	2	1:12
38.	1 Smart-board / Smart TV / Projector (with screen)	Where available	1	1:25
39.	Whiteboard/Chalkboard	4 X 8 Feet	1	1:25

MODULE VI

Unit Catego	Unit Code	Unit Name	Unit Duration (Hours)
CORE	0613 551 04 A	MOBILE APPLICATION DEVELOPMENT	270
BASIC	0031 541 01A	COMMUNICATION SKILLS	40
		Total	**310**

MOBILE APPLICATION DEVELOPMENT

ISCED UNIT CODE: 0613 551 04 A

TVET CDACC UNIT CODE: ICT/CU/SD/CR/03/6/MA

Relationship to Occupational Standards

This unit addresses the Unit of Competency: Develop Mobile Application

Duration of Unit: 270 Hours

Unit Description

This unit covers the competencies required to develop mobile application. It involves designing mobile application, writing mobile application source code, debugging mobile application, testing mobile application and publishing mobile application.

Summary of Learning Outcomes

Learning Outcomes	Duration (Hours)
1. Design Mobile Application	80
2. Write mobile application source code	110
3. Debug mobile application	40
4. Test mobile application	30
5. Publish mobile application	10
TOTAL	**270**

Learning Outcomes, Content and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Design Mobile Application	1.1. Design principles and Guidelines 1.1.1. Operating system 1.1.2. Consistency	● Practical Activities ● Project work ● Demonstration ● Group Work

	1.1.3. Device 1.1.4. Scalability 1.1.5. Simplicity 1.1.6. Mobile application design tools 1.1.6.1. Types of mobile application design tools e.g. Figma, Sketch, Adobe XD, InVision, Marvel 1.2. Mobile application functionality 1.2.1. Ease of Use (Usability) 1.2.2. Performance 1.2.3. Security 1.2.4. Compatibility 1.2.5. Offline Access 1.3. Mobile application interface 1.4. Mobile application output design	● Observation ● Third Party report ● Portfolio of Evidence ● Written tests
2. Write mobile application source code	2.1. Mobile application development tools 2.1.1. Integrated Development Environment (IDE) 2.1.2. Graphic User Interface (2.1.3. Emulator 2.1.4. Mobile SDK 2.2. Configure mobile application development environment 2.2.1. Android Studio 2.2.2. Xcode 2.2.3. Flutter 2.2.4. React Native 2.2.5. Visual Studio with Xamarin	● Practical Activities ● Project work ● Demonstration ● Group Work ● Observation ● Third Party report ● Portfolio of Evidence Written tests

	2.3. Mobile application interface 2.3.1. Setup 2.3.2. Develop 2.3.3. Test and Debug 2.3.4. Publish 2.4. Mobile application functionality 2.4.1. User Authentication 2.4.2. Navigation 2.4.3. Data Input and Management 2.4.4. Notifications and Alerts 2.4.5. Search Functionality 2.5. Integrate backend with frontend 2.5.1. Define API Endpoints 2.5.2. Set Up the Backend Server 2.5.3. Implement Database Models 2.5.4. Develop API Logic 2.5.5. Connect Frontend to API 2.5.6. Manage State in Frontend 2.5.7. Handle Error Responses 2.5.8. Test the Integration 2.6. Integrate Mobile application components with APIs 2.6.1. Define Component Requirements 2.6.2. Design API Endpoints 2.6.3. Implement API Calls in Components 2.6.4. Manage Component State 2.6.5. Handle Errors and Loading States 2.6.6. Data Binding	

	2.6.7. Testing API Integration 2.6.8. Monitor API Performance 2.6.9. Iterate Based on Feedback 2.6.10. Documentation 2.7. Workplace safety and health practices 2.7.1. Observe health safety practices 2.8. Methods of e-waste storage and disposal 2.8.1. Recycling 2.8.2. Landfilling 2.8.3. Incineration 2.8.4. Refurbishing 2.8.5. Donation 2.8.6. Component Harvesting 2.8.7. Safe Storage 2.8.8. Exporting for Recycling 2.8.9. Electronic Waste Collection Events 2.8.10. Manufacturer Take-Back Programs 2.9. E-waste management is demonstration 2.9.1. Understanding E-waste 2.9.2. E-waste Collection Methods 2.9.3. E-waste Sorting and Processing 2.9.4. Recycling Techniques 2.9.5. Responsible Disposal Practices	

3. Debug mobile application	3.1. Checking of mobile application source code 3.1.1. Error handling techniques 3.1.2. Source code revision 3.2. Debugging tools-bring from range 3.2.1. Integrated Development Environment (IDE) Debuggers 3.2.2. Print Statements 3.2.3. Profiling Tools 3.2.4. Memory Debuggers 3.2.5. Browser Developer Tools 3.2.6. Static Code Analysis Tools 3.2.7. Remote Debugging Tools 3.3. Regression testing 3.4. Prepare debugging report.	• Practical Activities • Project work • Demonstration • Group Work • Observation • Third Party report • Portfolio of Evidence Written tests
4. Test mobile application	4.1. Mobile application testing 4.1.1. Unit test 4.1.2. Integration test 4.1.3. Usability test 4.1.4. System testing 4.1.5. Security test 4.1.6. Performance test 4.1.7. Compatibility test 4.2. Mobile application test plan 4.2.1. Test environmental 4.2.2. Test scope 4.2.3. Schedule 4.3. Mobile application testing tools 4.3.1. Performance testing tools 4.3.2. functional testing tools 4.3.3. security testing tools 4.3.4. Cross-browser testing 4.3.5. Mobile-web application	• Practical Activities • Project work • Demonstration • Group Work • Observation • Third Party report • Portfolio of Evidence Written tests

	testing tools 4.3.6. Usability testing tools e,g. Google analytics 4.4. Test report preparation 4.4.1. Types of Test reports	
5. Publish mobile application	5.1. Identification of mobile publishing tools 5.1.1. Adobe InDesign 5.1.2. Canva 5.1.3. Lucidpress 5.1.4. Joomag 5.1.5. FlipHTML5 5.2. Generation of mobile application bundle 5.2.1. Android App Bundle (AAB) 5.2.2. Universal APK 5.2.3. iOS App Bundle (IPA) 5.2.4. Flutter App Bundle 5.2.5. React Native Bundle 5.2.6. Cordova/PhoneGap Bundle 5.3. Mobile application published 5.3.1. Application distribution through application stores	• Practical Activities • Project work • Demonstration • Group Work • Observation • Third Party report • Portfolio of Evidence Written tests

Suggested Delivery Methods

- Instructor led facilitation using active learning strategies
- Demonstration by trainer
- Practical work by trainees
- Viewing of related videos
- Field Visits
- Group Work

- Role plays
- Group projects

Recommended Resources for 25 Trainees

S/No.	Category/Item	Description/ Specifications	Quantity	Recommended Ratio (Item: Trainee)
A	**Learning Materials**			
40.	Internet connection	✓ For each computer	1	1:1
41.	Flip charts	A1	1	1:25
42.	Textbooks	For reference	3	3:25
B	**Learning Facilities & infrastructure**			
43.	Computer Laboratory	To accommodate 25 Learners	1	1:25
44.	Theory Room	furnished with 25 Arm-chairs and a suitable trainer's table	1	1:25
C	**Consumable materials**			
45.	Printing papers	A4	2 Reams	1:12
46.	Toner / Ink bottles	For printers	2 pcs	1:12
47.	White board markers	Assorted colors	20	4:5
D	**Tools and Equipment**			
48.	Computers	✓ Genuine Windows/Linux ✓ Genuine Microsoft office Software ✓ Google Workspace Account ✓ Antivirus Software ✓ Suitable IDE	25	1:1
49.	External storage media	HDD / SSD / Flash	1	1:25
50.	Printer	Working printer	2	1:12
51.	1 Smart-board / Smart TV / Projector (with screen)	Where available	1	1:25
52.	Whiteboard/Chalkboard	4 X 8 Feet	1	1:25

COMMUNICATION SKILLS

ISCED UNIT CODE: 0031 541 01A

TVET CDACC UNIT CODE: ICT/CU/SD/BC/01/6/MA

Relationship to Occupational Standards

This unit addresses the Unit of Competency: Apply Communication Skills

Duration of Unit: 40 hours

Unit Description

This unit covers the competencies required to apply communication skills. It involves applying communication channels, written, non-verbal, oral, and group communication skills.

Summary of Learning Outcomes

Learning Outcomes	Duration (Hours)
1. Apply communication channels	10
2. Apply written communication skills	12
3. Apply non-verbal communication skills	4
4. Apply oral communication skills	4
5. Apply group discussion skills	10
TOTAL	**40**

Learning Outcomes, Content, and Suggested Assessment Methods

Learning Outcome	Content	Suggested Assessment Methods
1. Apply communication channels	1.1 Communication process 1.1.1 Principles of effective communication 1.2 Channels/medium/modes of communication	• Oral questions • Written assessment • Observation • Portfolio of Evidence • Practical assessment

Learning Outcome	Content	Suggested Assessment Methods
	1.1.1 Factors to consider when selecting a channel of communication 1.1.2 Barriers to effective communication 1.2 Flow/patterns of communication 1.2.1 Sources of information 1.2.2 Organizational policies	• Third party report
2. Apply written communication skills	2.1 Types of written communication 2.2 Elements of communication 2.3 Organization requirements for written communication	• Oral assessment • Written assessment • Observation • Portfolio of Evidence • Practical assessment • Third party report
3. Apply non-verbal communication skills	3.1 Utilize body language and gestures 3.2 Apply body posture 3.3 Apply workplace dressing code	• Oral assessment • Written assessment • Observation • Portfolio of Evidence • Practical assessment • Third party report
4. Apply oral communication skills	4.1 Types of oral communication pathways 4.2 Effective questioning techniques 4.3 Workplace etiquette 4.4 Active listening	• Oral assessment • Written assessment • Observation • Portfolio of Evidence • Practical assessment • Third party report

Learning Outcome	Content	Suggested Assessment Methods
5. Apply group discussion skills	5.1 Establishing rapport 5.2 Facilitating resolution of issues 5.3 Developing action plans 5.4 Group organization techniques 5.5 Turn-taking techniques 5.6 Conflict resolution techniques 5.7 Team-work	• Oral assessment • Written assessment • Observation • Portfolio of Evidence • Practical assessment

Suggested Methods of Instruction

- Roleplaying
- Simulation
- Field trips
- Viewing of related videos
- Demonstrations
- Online Training
- Group Work.
- Instructor led facilitation using active learning strategies.

Recommended Resources for 25 trainees

S/No.	Category/Item	Description/ Specifications	Quantity	Recommended Ratio (Trainee: Item)
A	**Learning Materials**			
1.	Textbooks		5 pcs	5:1
2.	PowerPoint presentations	For trainer's use		
3.	Assorted colour of whiteboard markers	For trainer's use	2 packets	
4.	e-Didactics	For trainer's use		
5.	Flashcards			
6.	Flip charts			
7.	Whiteboard			

B	**Learning Facilities & infrastructure**			
8.	Lecture/theory room		1	25:1
C	**Consumable materials**			
9.	Printing Papers		1 ream	1:20
10.	Toners		2 pcs	13:1
11.	Internet			
D	**Tools and Equipment**			
12.	Projectors		1	25:1
13.	Printers		4	6:1
14.	Computers/Smartphones		25 pcs	1:1