

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

These are the specializations and their pre-requisites. These lists should be used as reference for curriculum maps.

AGRI-FISHERY ARTS

	Specialization	Number of Hours	Pre-requisite
1.	Agricultural Crops Production (NC I)	320 hours	
2.	Agricultural Crops Production (NC II) <i>updated based on TESDA Training Regulations published December 28, 2013</i>	640 hours	
3.	Agricultural Crops Production (NC III)	640 hours	Agricultural Crops Production (NC II)
4.	Animal Health Care Management (NC III)	320 hours	Animal Production (Poultry-Chicken) (NC II) or Animal Production (Ruminants) (NC II) or Animal Production (Swine) (NC II)
5.	Animal Production (Poultry-Chicken) (NC II) <i>updated based on TESDA Training Regulations published December 28, 2013</i>	320 hours	
6.	Animal Production (Large Ruminants) (NC II) <i>updated based on TESDA Training Regulations published December 28, 2013</i>	320 hours	
7.	Animal Production (Swine) (NC II) <i>updated based on TESDA Training Regulations published December 28, 2013</i>	320 hours	
8.	Aquaculture (NC II)	640 hours	
9.	Artificial Insemination (Large Ruminants) (NC II)	160 hours	Animal Production (Large Ruminants) (NC II)
10.	Artificial Insemination (Swine) (NC II)	160 hours	Animal Production (Swine) (NC II)
11.	Fish Capture (NC II)	640 hours	
12.	Fishing Gear Repair and Maintenance (NC III)	320 hours	
13.	Fish-Products Packaging (NC II)	320 hours	
14.	Fish Wharf Operation (NC I)	160 hours	
15.	Food Processing (NC II)	640 hours	
16.	Horticulture (NC III)	640 hours	Agricultural Crops Production (NC II)
17.	Landscape Installation and Maintenance (NC II)	320 hours	
18.	Organic Agriculture (NC II)	320 hours	
19.	Pest Management (NC II)	320 hours	
20.	Rice Machinery Operations (NC II)	320 hours	
21.	Rubber Processing (NC II)	320 hours	
22.	Rubber Production (NC II)	320 hours	
23.	Slaughtering Operations (Hog/Swine/Pig) (NC II)	160 hours	

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

HOME ECONOMICS

	Specialization	Number of Hours	Pre-requisite
1.	Attractions and Theme Parks Operations with Ecotourism (NC II)	160 hours	
2.	Barbering (NC II)	320 hours	
3.	Bartending (NC II)	320 hours	
4.	Beauty/Nail Care (NC II)	160 hours	
5.	Bread and Pastry Production (NC II)	160 hours	
6.	Caregiving (NC II)	640 hours	
7.	Commercial Cooking (NC III)	320 hours	Cookery (NC II)
8.	Cookery (NC II)	320 hours	
9.	Dressmaking (NC II)	320 hours	
10.	Events Management Services (NC III)	320 hours	
11.	Fashion Design (Apparel) (NC III)	640 hours	Dressmaking (NC II) or Tailoring (NC II)
12.	Food and Beverage Services (NC II) <i>updated based on TESDA Training Regulations published December 28, 2013</i>	160 hours	
13.	Front Office Services (NC II)	160 hours	
14.	Hairdressing (NC II)	320 hours	
15.	Hairdressing (NC III)	640 hours	Hairdressing (NC II)
16.	Handicraft (Basketry, Macrame) (Non-NC)	160 hours	
17.	Handicraft (Fashion Accessories, Paper Craft) (Non-NC)	160 hours	
18.	Handicraft (Needlecraft) (Non-NC)	160 hours	
19.	Handicraft (Woodcraft, Leathercraft) (Non-NC)	160 hours	
20.	Housekeeping (NC II) <i>updated based on TESDA Training Regulations published December 28, 2013</i>	160 hours	
21.	Local Guiding Services (NC II)	160 hours	
22.	Tailoring (NC II)	320 hours	
23.	Tourism Promotion Services (NC II)	160 hours	
24.	Travel Services (NC II)	160 hours	
25.	Wellness Massage (NC II)	160 hours	

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

INDUSTRIAL ARTS

	Specialization	Number of Hours	Pre-requisite
1.	Automotive Servicing (NC I) <i>updated based on TESDA Training Regulations published December 28, 2013</i>	640 hours	
2.	Automotive Servicing (NC II)	640 hours	Automotive Servicing (NC I)
3.	Carpentry (NC II)	640 hours	
4.	Carpentry (NC III)	320 hours	Carpentry (NC II)
5.	Construction Painting (NC II)	160 hours	
6.	Domestic Refrigeration and Air-conditioning (DOMRAC) Servicing (NC II)	640 hours	
7.	Driving (NC II)	160 hours	
8.	Electrical Installation and Maintenance (NC II)	640 hours	
9.	Electric Power Distribution Line Construction (NC II)	320 hours	Electrical Installation and Maintenance (NC II)
10.	Electronic Products Assembly and Servicing (NC II) <i>updated based on TESDA Training Regulations published December 28, 2013</i>	640 hours	
11.	Furniture Making (Finishing) (NC II)	640 hours	
12.	Instrumentation and Control Servicing (NC II)	320 hours	Electronic Products Assembly and Servicing (EPAS) (NC II)
13.	Gas Metal Arc Welding (GMAW) (NC II)	320 hours	Shielded Metal Arc Welding (SMAW) (NC II)
14.	Gas Tungsten Arc Welding (GTAW) (NC II)	320 hours	Shielded Metal Arc Welding (GMAW) (NC II)
15.	Machining (NC I)	640 hours	
16.	Machining (NC II)	640 hours	Machining (NC I)
17.	Masonry (NC II)	320 hours	
18.	Mechatronics Servicing (NC II)	320 hours	Electronic Products Assembly and Servicing (EPAS) (NC II)
19.	Motorcycle/Small Engine Servicing (NC II)	320 hours	
20.	Plumbing (NC I)	320 hours	
21.	Plumbing (NC II)	320 hours	Plumbing (NC I)
22.	Refrigeration and Air-Conditioning (Packaged Air-Conditioning Unit [PACU]/Commercial Refrigeration Equipment [CRE]) Servicing (NC III)	640 hours	Domestic Refrigeration and Air-conditioning (DOMRAC) Servicing (NC II)
23.	Shielded Metal Arc Welding (NC I)	320 hours	
24.	Shielded Metal Arc Welding (NC II)	320 hours	Shielded Metal Arc Welding (NC I)
25.	Tile Setting (NC II)	320 hours	
26.	Transmission Line Installation and Maintenance (NC II)	640 hours	Electrical Installation and Maintenance (NC II)

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

INFORMATION, COMMUNICATIONS AND TECHNOLOGY (ICT)

	Specialization	Number of Hours	Pre-requisite
1.	Animation (NC II)	320 hours	
2.	Broadband Installation (Fixed Wireless Systems) (NC II)	160 hours	Computer Systems Servicing (NC II)
3.	Computer Programming (.Net Technology) (NC III) <i>updated based on TESDA Training Regulations published December 28, 2013</i>	320 hours	
4.	Computer Programming (Java) (NC III) <i>updated based on TESDA Training Regulations published December 28, 2013</i>	320 hours	
5.	Computer Programming (Oracle Database) (NC III) <i>updated based on TESDA Training Regulations published December 28, 2013</i>	320 hours	
6.	Computer Systems Servicing (NC II) <i>updated based on TESDA Training Regulations published December 28, 2007</i>	640 hours	
7.	Contact Center Services (NC II)	320 hours	
8.	Illustration (NC II)	320 hours	
9.	Medical Transcription (NC II)	320 hours	
10.	Technical Drafting (NC II)	320 hours	
11.	Telecom OSP and Subscriber Line Installation (Copper Cable/POTS and DSL) (NC II)	320 hours	Computer Systems Servicing (NC II)
12.	Telecom OSP Installation (Fiber Optic Cable) (NC II)	160 hours	Computer Systems Servicing (NC II)

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

Course Description:

This is a specialization course that leads to a Programming (Java) National Certificate Level III (NC III). It discusses six (6) basic competencies, two (2) common competencies, and two (2) core competencies that a Grade 11 Technical-Vocational Education (TVE) student ought to possess.

The basic competencies are for learners to: 1) lead workplace communication, 2) lead small teams, 3) develop and practice negotiation skills, 4) solve problems related to work activities, 5) use mathematical concepts and techniques, and 6) use relevant technologies. The common competencies are for learners to: 1) apply quality standards and 2) perform computer operations. The core competencies are for learners to: 1) perform object-oriented analysis and design in Java technology, and 2) create and fine-tune Java technology applications using object-oriented programming concept.

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
Introduction 1. Relevance of the course 2. Concepts and core competencies in Programming (Java)	The learners demonstrate an understanding of key concepts, underlying principles, and core competencies in Programming (Java)	The learners shall be able to independently create/ provide quality and marketable product and/ or service in Programming (Java), as prescribed by TESDA Training Regulations	<i>The learners...</i> 1. Discuss the relevance of the course 2. Explain key concepts of common competencies 3. Explain core competencies of Programming (Java)	
BASIC COMPETENCIES				
LESSON 1: LEAD WORKPLACE COMMUNICATION (LWC)				
<ul style="list-style-type: none"> • Organization requirements for written and electronic communication methods • Effective verbal communication methods • Organizing information • Understanding and conveying intended meaning • Participation in various workplace discussions • Complying with organization requirements for the use of written and electronic communication methods 	The learners demonstrate an understanding of the principles and concepts in leading in the dissemination and discussion of ideas, information, and issues in the workplace	The learners independently lead in the dissemination and discussion of ideas, information, and issues in the workplace based on TESDA Training Regulations	LO 1. Communicate information about workplace processes 1.1 Select appropriate communication method 1.2 Communicate multiple operations involving several topic areas accordingly 1.3 Use questions to gain extra information 1.4 Identify correct sources of information 1.5 Select and organize information correctly 1.6 Undertake verbal and written reporting when required 1.7 Maintain communication skills in all situations	TLE_ICTJAVA1-12LWC-Ia-1

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
<ul style="list-style-type: none"> • Effective verbal communication methods in the workplace • Understanding and conveying intended meaning • Participation in various workplace discussions • Complying with organization requirements for the use of written and electronic communication methods 			<p>LO 2. Lead workplace discussions</p> <p>2.1 Seek responses to workplace issues</p> <p>2.2 Respond immediately to workplace issues</p> <p>2.3 Make constructive contributions to workplace discussions on such issues as production, quality, and safety</p> <p>2.4 Communicate goals/objectives and action plan undertaken in the workplace</p>	<p>TLE_ICTJAVA1 1-12LWC-Ia</p>
<ul style="list-style-type: none"> • Effective verbal communication methods • Organizing information • Understand and convey intended meaning • Participation in various workplace discussions • Comply with organization requirements for the use of written and electronic communication methods 			<p>LO 3. Identify and communicate issues arising in the workplace</p> <p>3.1 Identify issues and problems as they arise</p> <p>3.2 Organize information regarding problems and issues coherently to ensure clear and effective communication</p> <p>3.3 Initiate dialogue with appropriate personnel</p> <p>3.4 Raise communication problems and issues as they arise</p>	<p>TLE_ICTJAVA1 1-12LWC-Ia-3</p>
LESSON 2: LEAD SMALL TEAMS (LST)				
<ul style="list-style-type: none"> • Company policies and procedures • Relevant legal requirements • How performance expectations are set • Methods of monitoring performance • Client expectations • Team member’s duties and responsibilities • Communication skills required for leading teams • Informal performance counseling skills • Team-building skills • Negotiating skills 	<p>The learners demonstrate an understanding of the principles and concepts in leading small teams, including setting and maintaining team and individual performance standards</p>	<p>The learners independently lead small teams, including setting and maintaining team and individual performance standards based on TESDA Training Regulations</p>	<p>LO 1. Provide team leadership</p> <p>1.1 Identify and present work requirements to team members</p> <p>1.2 Communicate reasons for instructions and requirements to team members</p> <p>1.3 Recognize, discuss, and deal with team members’ queries and concerns</p>	<p>TLE_ICTJAVA1 1-12LST-Ia-4</p>

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
<ul style="list-style-type: none"> • How performance expectations are set • Methods of monitoring performance • Client expectations • Team member’s duties and responsibilities • Communication skills required for leading teams • Informal performance counseling skills 			<p>LO 2. Assign responsibilities among members</p> <p>2.1 Allocate duties and responsibilities with regard to the skills, knowledge, and aptitude required to properly undertake the assigned task and according to company policy</p> <p>2.2 Allocate duties with regard to individual preference, and domestic and personal considerations, whenever possible</p>	<p>TLE_ICTJAVA1 1-12LST-Ib-5</p>
<ul style="list-style-type: none"> • How performance expectations are set • Methods of monitoring performance • Client expectations • Team member’s duties and responsibilities • Informal performance counseling skills 			<p>LO 3. Set performance expectation for team members</p> <p>3.1 Establish performance expectations based on client needs and according to assignment requirements</p> <p>3.2 Base performance expectations on individual team member’s duties and area of responsibility</p> <p>3.3 Discuss performance expectations and disseminate to individual team members</p>	<p>TLE_ICTJAVA1 1-12LST-Ib-6</p>
<ul style="list-style-type: none"> • How performance expectations are set • Methods of monitoring performance • Team member’s duties and responsibilities • Communication skills required for leading teams • Informal performance counseling skills • Team-building skills • Negotiating skills 			<p>LO 4. Supervise team performance</p> <p>4.1 Monitor performance against defined performance criteria and/or assignment instructions and corrective action taken if required</p> <p>4.2 Provide team members with feedback, positive support, and advice on strategies to overcome deficiencies</p> <p>4.3 Refer to appropriate personnel according to employer policy when performance issues cannot be rectified or addressed within the team to appropriate personnel</p> <p>4.4 Keep team members informed of any changes in the priority allocated to</p>	<p>TLE_ICTJAVA1 1-12LST-Ib-7</p>

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
			assignments or tasks, which might impact on client/customer needs and satisfaction 4.5 Monitor operations to ensure that employer/client needs and requirements are met 4.6 Provide follow-up communication on all issues affecting the team 4.7 Complete all relevant documentation in accordance with company procedures	
LESSON 3: DEVELOP AND PRACTICE NEGOTIATION SKILLS (DPN)				
<ul style="list-style-type: none"> • Codes of practice and guidelines for the organization • Organizations policy and procedures for negotiations • Decision making procedures and conflict-resolution strategies • Problem-solving strategies to deal with unexpected questions and attitudes during negotiation • Interpersonal skills to develop rapport with other parties • Communication skills (verbal and listening) 	The learners demonstrate an understanding of the principles and concepts in planning and participating in negotiations	The learners independently plan and participate in negotiations based on TESDA Training Regulations	LO 1. Plan negotiations. 1.1 Identify and include information on preparing for negotiation in the plan. 1.2 Identify and include information on creating non-verbal environments for positive negotiating in the plan. 1.3 Identify and include information on active listening in the plan 1.4 Identify and include information on different questioning techniques in the plan 1.5 Check information to ensure it is correct and up-to- date	TLE ICTJAVA1 1-12DPN-Ib-c-8
<ul style="list-style-type: none"> • Codes of practice and guidelines for the organization • Organizations policy and procedures for negotiations • Decision-making procedures and conflict- resolution strategies • Problem-solving strategies to deal with unexpected questions and attitudes during negotiation • Interpersonal skills to develop rapport with other parties 			LO 2. Participate in negotiations 2.1 Agree on criteria for successful outcome by all parties 2.2 Consider desired outcome of all parties 2.3 Use appropriate language throughout the negotiation 2.4 Use a variety of questioning techniques 2.5 Documentation and agreement on the issues and processes by all parties 2.6 Discuss possible solutions and their viability 2.7 Confirm and record areas for agreement 2.8 Agree on follow-up action by all parties	TLE ICTJAVA1 1-12DPN-Ic-9

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
LESSON 4: SOLVE PROBLEM RELATED TO WORK ACTIVITIES (PRW)				
<ul style="list-style-type: none"> • Competence includes a thorough knowledge and understanding of the process, normal operating parameters, and product quality to recognize nonstandard situations • Competence to include the ability to apply and explain, sufficient for the identification of fundamental cause, determining the corrective action and provision of recommendations <ul style="list-style-type: none"> - Relevant equipment and operational processes - Enterprise goals, targets and measures - Enterprise quality, OHS and environmental requirement - Principles of decision making strategies and techniques - Enterprise information systems and data collation - Industry codes and standards • Using range of formal problem-solving techniques • Identifying and clarifying the nature of the problem • Devising the best solution • Evaluating the solution • Implementing a developed plan to rectify the problem 	<p>The learners demonstrate an understanding of the principles and concepts in solving problems in the workplace, including the application of problem-solving techniques, and in determining and resolving the root cause of problems</p>	<p>The learners independently solve problems in the workplace, including the application of problem-solving techniques, and determine and resolve the root cause of problems based on TESDA Training Regulations</p>	<p>LO 1. Identify the problem</p> <ul style="list-style-type: none"> 1.1 Identify variances from normal operating parameters and product quality 1.2 Define extent, cause, and nature of the problem through observation, investigation, and analytical techniques 1.3 State and specify problems clearly 	<p>TLE_ ICTJAVA1 1-12PRW-Ic- 10</p>
<ul style="list-style-type: none"> • Using a range of formal problem-solving techniques • Identifying and clarifying the nature of the problem 			<p>LO 2. Determine fundamental causes of the problem</p> <ul style="list-style-type: none"> 2.1 Identify possible causes based on experience and the use of problem-solving 	<p>TLE_ ICTJAVA1 1-12-PRW-Ic- 11</p>

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
<ul style="list-style-type: none"> Devising the best solution Evaluating the solution Implementing a developed plan to rectify the problem 			tools/ analytical techniques. 2.2 Develop possible cause statements based on findings 2.3 Identify fundamental causes per results of investigation conducted	
<ul style="list-style-type: none"> Using a range of formal problem-solving techniques Identifying and clarifying the nature of the problem Devising the best solution Evaluating the solution Implementing a developed plan to rectify the problem 			LO 3. Determine corrective action 3.1 Consider all possible options for resolution of the problem 3.2 Consider strengths and weaknesses of possible options 3.3 Determine corrective actions to resolve the problem and possible future causes 3.4 Develop action plans identifying measurable objectives, resource needs, and timelines in accordance with safety and operating procedures	TLE_ICTJAVA1 1-12-PRW-Id-12
<ul style="list-style-type: none"> Identifying and clarifying the nature of the problem Devising the best solution Evaluating the solution Implementing a developed plan to rectify the problem 			LO 4. Provide recommendations to manager 4.1 Prepare reports on recommendations 4.2 Present recommendations to appropriate personnel 4.3 Follow up recommendations, if required	TLE_ICTJAVA1 1-12PRW-Id-13
LESSON 5: USE MATHEMATICAL CONCEPTS AND TECHNIQUES (MCT)				
<ul style="list-style-type: none"> Fundamental operation (addition, subtraction, division, multiplication) Measurement system Precision and accuracy Basic measuring tools/devices 	The learners demonstrate an understanding of the principles and concepts in selecting, sourcing, and applying appropriate and affordable technologies in the workplace	The learners independently select, source, and apply appropriate and affordable technologies in the workplace based on TESDA Training Regulations	LO 1. Identify mathematical tools and techniques to solve problems 1.1 Identify problem areas based on the given condition 1.2 Select mathematical techniques based on the given problem	TLE_ICTJAVA1 1-12MCT-Id-14
<ul style="list-style-type: none"> Measurement system Precision and accuracy Basic measuring tools/devices Applying mathematical 			LO 2. Apply mathematical procedure/ solution 2.1 Apply mathematical techniques based on the problem identified	TLE_ICTJAVA1 1-12MCT-Id-15

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
computations <ul style="list-style-type: none"> • Using calculator • Using different measuring tools 			2.2 Perform mathematical computations to the level of accuracy required for the problem 2.3 Determine results of mathematical computation and verify based on job requirements	
<ul style="list-style-type: none"> • Precision and accuracy • Applying mathematical computations • Using calculator • Using different measuring tools 			LO 3. Analyze results 1.1 Identify problem areas based on given condition 1.2 Apply appropriate action in case of error	TLE_ICTJAVA1 1-12MCT-Ie-16
LESSON 6: USE RELEVANT TECHNOLOGIES (URT)				
<ul style="list-style-type: none"> • Awareness of technology and its functions • Repair and maintenance procedure • Operating instructions • Applicable software • Communication techniques • Health and safety procedures • Company policy in relation to relevant technology • Different management concepts • Technology adaptability 	The learners demonstrate an understanding of the principles in applying mathematical concepts and techniques	The learners independently apply mathematical concepts and techniques based on TESDA Training Regulations	LO 1. Study/Select appropriate technologies 1.1. Determine usage of different technologies based on job requirements 1.2. Select appropriate technology as per work specification	TLE_ICTJAVA1 1-12URT-Ie-17
<ul style="list-style-type: none"> • Repair and maintenance procedure • Operating instructions • Applicable software • Communication techniques • Health and safety procedure • Company policy in relation to relevant technology • Different management concepts • Technology adaptability • Relevant technology application/ 			LO 2. Apply relevant technologies 2.1 Use relevant technology effectively in carrying out function 2.2 Use applicable software and hardware as per task requirement 2.3 Observe and practice management concepts as per established industry practice	TLE_ICTJAVA1 1-12URT-Ie-18

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
implementation <ul style="list-style-type: none"> • Basic communication skills • Software applications skills 				
<ul style="list-style-type: none"> • Relevant technology application/ implementation • Basic communication skills • Software applications skills • Basic troubleshooting skills 			LO 3. Maintain/enhance relevant technology 3.1 Perform maintenance of technology in accordance with industry-standard operating procedure, manufacturer’s operating guidelines, and occupational health and safety procedure to ensure operative ability 3.2 Update technology through continuing education or training in accordance with job requirement 3.3 Report immediately technology failure/ defect to the concerned person or section for appropriate action.	TLE_ ICTJAVA1 1-12URT-Ie-19
COMMON COMPETENCIES				
LESSON 7: APPLY QUALITY STANDARDS (AQS)				
<ul style="list-style-type: none"> • Relevant production processes, materials, and products • Characteristics of materials, software, and hardware used in production processes • Quality-checking procedures • Workplace procedures • Safety and environmental aspects of production processes • Fault identification and reporting • Quality improvement processes • Reading skills required to interpret work instruction • Communication skills needed to interpret and apply defined work procedures • Carry out work in accordance with OHS policies and procedures • Critical thinking 	The learners demonstrate an understanding of the principles and concepts in applying quality standards	The learners independently apply quality standards based on TESDA Training Regulations	LO 1. Assess quality of received materials 1.1 Obtain work instruction and carry out work in accordance with standard operating procedures. 1.2 Check and receive materials against workplace standards and specifications 1.3 Identify and isolate faulty materials related to work 1.4 Record and/or report faults and identified causes to the supervisor concerned in accordance with workplace procedures 1.5 Replace faulty materials in accordance with workplace procedures	TLE_ ICTJAVA1 1-12-AQS-If-20

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
<ul style="list-style-type: none"> • Relevant production processes, materials, and products • Characteristics of materials, software, and hardware used in production processes • Quality-checking procedures • Workplace procedures • Safety and environmental aspects of production processes • Fault identification and reporting • Quality improvement processes • Reading skills required to interpret work instruction • Communication skills needed to interpret and apply defined work procedures • Carry out work in accordance with OHS policies and procedures • Critical thinking • Solution providing and decision making 			<p>LO 2. Assess own work</p> <p>2.1 Identify and use documentation relative to quality within the company</p> <p>2.2 Check completed work against workplace standards relevant to the task undertaken</p> <p>2.3 Identify and isolate errors</p> <p>2.4 Record information on the quality and other indicators of production performance in accordance with workplace procedures</p> <p>2.5 Document and report causes in cases of deviations from specific quality standards in accordance with the workplace’s standard operating procedures.</p>	<p>TLE_ICTJAVA1 1-12AQS-Ig- 21</p>
<ul style="list-style-type: none"> • Relevant production processes, materials, and products • Characteristics of materials, software, and hardware used in production processes • Quality-checking procedures • Workplace procedures • Safety and environmental aspects of production processes • Fault identification and reporting • Quality improvement processes • Reading skills required to interpret work instruction • Communication skills needed to 			<p>LO 3. Engage in quality improvement</p> <p>3.1 Participate in process improvement procedures relative to workplace assignment</p> <p>3.2 Carry out work in accordance with process improvement procedures</p> <p>3.3 Monitor performance of operation or quality of product of service to ensure customer satisfaction</p>	<p>TLE_ICTJAVA1 1-12AQS-Ih- 22</p>

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
interpret and apply defined work procedures <ul style="list-style-type: none"> • Carry out work in accordance with OHS policies and procedures • Critical thinking • Solution providing and decision making 				
LESSON 8: PERFORM COMPUTER OPERATIONS (PCO)				
<ul style="list-style-type: none"> • Reading and comprehension skills required to interpret work instructions and to interpret basic user manuals • OH and S principles and responsibilities • Main types of computers and basic features of different operating systems • Main parts of a computer • Storage devices and basic categories of memory • Relevant types of software • General security, privacy legislation, and copyright 	The learners demonstrate an understanding of the principles and concepts in performing computer operations	The learners independently perform computer operations based on TESDA Training Regulations	LO 1. Plan and prepare for tasks at hand <ol style="list-style-type: none"> 1.1 Determine the requirements of tasks in accordance with the required output 1.2 Select appropriate hardware and software according to task assigned and required outcome 1.3 Plan task to ensure that OH and S guidelines and procedures followed 1.4 Follow client-specific guidelines and procedures 1.5 Apply required data security guidelines in accordance with existing procedures 	TLE ICTJAVA1 1-12PCO-Ii-23
<ul style="list-style-type: none"> • Relevant types of software • Communication skills to identify lines of communication, request advice, follow instructions, and receive feedback • Storage devices and basic categories of memory • Basic ergonomics of keyboard and computer user 			LO 2. Input data into computer <ol style="list-style-type: none"> 2.1 Enter data into the computer using appropriate program/application in accordance with company procedure 2.2 Check accuracy of information and information saved in accordance with standard operating procedures 2.3 Store inputted data in storage media according to requirements 2.4 Perform work within ergonomic guidelines 	TLE ICTJAVA1 1-12PCO-Ij-24

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
<ul style="list-style-type: none"> • Relevant types of software • Business Application • System software • Basic ergonomics of keyboard and computer user 			<p>LO 3. Access information using computer</p> <p>3.1 Select correct program/application based on job requirements</p> <p>3.2 Access program/application containing the information required according to company procedures</p> <p>3.3 Select, open, and close desktop icons correctly for navigation purposes</p> <p>3.4 Carry out keyboard techniques in line with OHS requirements for safe use of keyboards</p>	<p>TLE_ICTJAVA1 1-12PCO-IIa-25</p>
<ul style="list-style-type: none"> • Relevant types of software • Computer peripherals • Storage devices and basic categories of memory 			<p>LO 4. Produce output/data using computer system</p> <p>4.1 Process entered data using appropriate software commands</p> <p>4.2 Print data as required using computer hardware/peripheral devices in accordance with standard operating procedures</p> <p>4.3 Transfer files and data between compatible systems using computer software, hardware/peripheral devices in accordance with standard operating procedures</p>	<p>TLE_ICTJAVAT 11-12PCO-IIb-c-26</p>
<ul style="list-style-type: none"> • Web surfing • Web browsers • Search engines • URLs and keywords • Links • Bookmarking 			<p>LO 5. Use basic functions of a web browser to locate information</p> <p>5.1 Establish information requirements for Internet search</p> <p>5.2 Launch browser</p> <p>5.3 Load search engine</p> <p>5.4 Enter appropriate search criteria and/or URL of site</p> <p>5.5 Follow relevant links to locate required information</p> <p>5.6 Bookmark and print useful pages as required</p>	<p>TLE_ICTJAVA1 1-12PCO-IIId-27</p>

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
<ul style="list-style-type: none"> Security measures Anti-virus software/programs File Management 			LO 6. Maintain computer equipment 6.1 Implement procedures for ensuring security of data, including regular back-ups and virus checks, in accordance with standard operating procedures 6.2 Implement basic file maintenance procedures in line with standard operating procedures	TLE_ICTJAVA1 1-12PCO-IIe-28
CORE COMPETENCIES				
LESSON 9: PERFORMING OBJECT-ORIENTED ANALYSIS AND DESIGN (POAD)				
<ul style="list-style-type: none"> Java technology and the Java programming language JAVA framework Basic computer operation skills Importing JAVA packages JAVA data types Uses of various Java programming language constructs to create several Java technology applications <ul style="list-style-type: none"> Define the scope of variables Define the structure of a Java class Differentiate between default and user-defined constructors Logic analysis Uses of decision and looping constructs and methods to dictate program flow 	The learners demonstrate an understanding of the principles and concepts in demonstrating knowledge of Java technology and Java programming language, and performing object-oriented analysis and design	The learners independently and effectively use object-oriented technologies and the use of software modeling, as applied to a software development process, and present one practical & complete object-oriented analysis and design (OOAD) roadmap based on TESDA Training Regulations	LO 1. Apply basics of Java language 1.1 Demonstrate knowledge of Java technology and Java programming 1.2 Create executable Java applications in accordance with Java framework 1.3 Import Java packages to make them accessible in the code 1.4 Demonstrate working with Java Data types in accordance with Java framework 1.5 Demonstrate using Operators and Decision Constructs in accordance with Java framework 1.6 Demonstrate creating and using Arrays in accordance with Java framework 1.7 Demonstrate using Loop Constructs in accordance with Java framework	TLE_ICTJAVA1 1-12POAD-IIf-i-29
<ul style="list-style-type: none"> Basic object oriented concepts such as inheritance, polymorphism, encapsulation, and abstraction Uses and manipulation of object references to write simple error handling code <ul style="list-style-type: none"> Differentiate among checked exceptions, Runtime 			LO 2. Work with inheritance and handling exceptions 1.1 Implement inheritance in accordance with Java framework 1.2 Develop code that demonstrates the use of polymorphism in accordance with Java framework 1.3 Use Super to access objects and constructors in accordance with Java	TLE_ICTJAVA1 1-12POAD-IIj-IIIa-c-30

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
<p>Exceptions, and Error</p> <ul style="list-style-type: none"> - Describe what exceptions are used for in Java - Invoke a method that throws an Exception - Recognize common exception classes and categories <p>• Code writing and debugging skills</p>			<p>framework</p> <p>1.4 Use abstract classes and interfaces in accordance with Java framework</p> <p>1.5 Determine how exceptions alter normal program flow by creating a try-catch block</p>	
<ul style="list-style-type: none"> • Fundamentals of Object Oriented (OO) terminologies • Concepts of object oriented • Understanding of basic object oriented concepts such as inheritance, encapsulation, and abstraction • Intermediate Java programming and object-oriented (OO) concepts in Java technology programs <ul style="list-style-type: none"> - Determine the effect upon object references and primitive values when they are passed into methods that change the values - Differentiate between the type of a reference and the type of an object; determine when casting is necessary 			<p>LO 3. Examine object-oriented concepts and terminology</p> <p>3.1 Describe important object-oriented (OO) concepts in accordance with Java framework</p> <p>3.2 Define fundamental OO terminology in accordance with Java framework</p>	<p>TLE_ICTJAVA1 1-12POAD- IIIId-g-31</p>
<ul style="list-style-type: none"> • Object-Oriented Software Development (OOSD) • Process of OOSD • Benefits of modeling software • OOSD workflows and disciplines 			<p>LO 4. Explain modeling and the software development process</p> <p>4.1 Explain Object-Oriented Software Development (OOSD) process in accordance with Java framework</p> <p>4.2 Explain benefits of modeling software in accordance with Java framework</p> <p>4.3 Explain purpose, activities, and artifacts of the following OOSD workflows (disciplines)</p>	<p>TLE_ICTJAVA1 1-12POAD- IIIh-j-IVa-32</p>

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
<ul style="list-style-type: none"> • Using CASE diagram • Developing CASE diagram for a software system • Setting the goals of developed CASE diagram • Creating and using the CASE forms • Recognize and document use case dependencies using UML notation for extends, includes, and generalization • Describe how to manage the complexity of Use Case Diagrams by creating UML packaged views; identify and document scenarios for a use case • Describe how to reference included and extending use cases • Identify and document nonfunctional requirements (NFRs), business rules, risks, and priorities for a use case 			<p>LO 5. Create Use Case diagrams and Use Case scenarios</p> <p>5.1 Justify the need for a Use Case Diagram in accordance with Java framework</p> <p>5.2 Develop Use Case Diagram for a software system based on the goals of the business owner</p> <p>5.3 Develop Use Case Diagrams based on the goals of all the stakeholders</p> <p>5.4 Create Use Case form describing a summary of the scenarios in the main and alternate flows</p>	<p>TLE_ICTJAVA1 1-12POAD- IVb-e-33</p>
<ul style="list-style-type: none"> • Purpose and elements of the design model • Essential elements of a UML Communication diagram • Creating the communication diagram view of the Design model • Creating the sequence diagram of the Design model • Managing the complexity of Use Case Diagrams by creating UML packaged views Identify and document scenarios for a use case 			<p>LO 6. Transition from analysis to design using interaction diagrams</p> <p>6.1 Explain purpose and elements of the Design model in accordance with Java framework</p> <p>6.2 Identify essential elements of a UML Communication diagram in accordance with Java framework</p> <p>6.3 Create communication diagram view of the Design model in accordance with Java framework</p> <p>6.4 6.4 Create sequence diagram view of the Design model in accordance with Java</p>	<p>TLE_ICTJAVA1 1-12POAD-IVf- i-34</p>

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
<ul style="list-style-type: none"> • Referencing included and extending use cases • Identify and document nonfunctional requirements (NFRs), business rules, risks, and priorities for a use case • Essential elements in an Activity diagram • Providing Supplementary Specification Document 			framework	
<ul style="list-style-type: none"> • Difference between architecture and design • Qualities of tiers and layers • Architecture workflow • Architecture Tiers • For Architecture purposes: <ul style="list-style-type: none"> - Managing the complexity of Use Case Diagrams by creating UML packaged views Identify and document scenarios for a use case - Referencing included and extending use cases - Documenting nonfunctional requirements (NFRs), business rules, risks, and priorities for a use case - Identify the purpose of a Supplementary Specification Document 			<p>LO 7. Introduce architectural concepts and architecture tiers diagrams</p> <p>7.1 Distinguish difference between architecture and design in accordance with Java framework</p> <p>7.2 Describe tiers, layers, and systemic qualities in accordance with Java framework</p> <p>7.3 Describe architecture workflow in accordance with Java framework</p> <p>7.4 Define architecture tiers in accordance with Java framework</p>	TLE ICTJAVA1 1-12POAD-IVj-Ia-c-35
<ul style="list-style-type: none"> • Methods with arguments and return values • Static keywords and its applications • Overload method and access modifiers 			<p>LO 8. Work with methods and encapsulation</p> <p>8.1 Create methods with arguments and return values in accordance with Java framework</p> <p>8.2 Apply static keywords to methods and fields in accordance with Java framework</p>	TLE ICTJAVA1 1-12POAD-Id-g-36

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
<ul style="list-style-type: none"> Principles of encapsulation and its application 			8.3 Create overloaded method in accordance with Java framework 8.4 Apply access modifiers in accordance with Java framework 8.5 Apply encapsulation principles to a class in accordance with Java framework	
LESSON 10: CREATE AND FINE TUNE JAVA TECHNOLOGY APPLICATION USING OBJECT-ORIENTED PROGRAM CONCEPTS (CFJA)				
<ul style="list-style-type: none"> Knowledge of Java I/O Fundamentals <ul style="list-style-type: none"> Read and write data from the console Use streams to read and write files Knowledge Java File I/O (NIO.2) <ul style="list-style-type: none"> Use the Path class to operate on file and directory paths Use the Files class to check, delete, copy, or move a file or directory Read and change file and directory attributes Recursively access a directory tree using the Directory Stream and File Visit or interfaces Find a class using the Path Matcher class Watch a directory for changes by using Watcher Service Accessing modifiers of JAVA frameworks Use of operators and castings Virtual method invocation and its uses Override methods from the Object class and its uses Package and import statements and its uses 	The learners demonstrate an understanding of the core Application Programming Interfaces (API) used to design object-oriented applications with Java and, how to write database programs with JDBC	The learners independently perform applying core Application Programming Interfaces (API) used to design object-oriented applications with Java and, how to write database programs with JDBC based on TESDA Training Regulations	LO 1. Apply basics of Java class design 1.1 Use access modifiers in accordance with Java framework 1.2 Use instance of operator and casting in accordance with Java framework 1.3 Use virtual method invocation in accordance with Java framework 1.4 Use override methods from the Object class to improve the functionality of class 1.5 Use package and import statements in accordance with Java framework	TLE_ICTJAVA1 1-12CFJA-Ih-j- IIa-37

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
<ul style="list-style-type: none"> • Identification and application of abstract classes • Construction of abstract JAVA classed and subclasses • Static and final keywords • Creating top level and nested classes • Using enumerated types • Using declares, implements, and/or extends interfaces • Application of Object Oriented Design Principles (OODP) • Principles of Generics and Collections 			<p>LO 2. Apply Java advance class design and object oriented design principles</p> <p>2.1 Identify proper use of when and how to apply abstract classes</p> <p>2.2 Construct abstract Java classes and subclasses in accordance with Java framework</p> <p>2.3 Use static and final keywords in accordance with Java framework</p> <p>2.4 Create top-level and nested classes in accordance with Java framework</p> <p>2.5 Use enumerated types in accordance with Java framework</p> <p>2.6 Write code that declares, implements, and/or extends interfaces</p> <p>2.7 Apply Object Oriented Design Principles in accordance with Java framework</p> <p>2.8 Use Generics and Collections Principles in accordance with Java framework</p>	<p>TLE_ ICTJAVA1 1-12CFJA-IIb- e-38</p>
<ul style="list-style-type: none"> • Using search, parse, and building strings • Formatting string and using throws statements • Using try statements with multi-catch • Using Autoclose features with a try-with-resources • Creating customize exceptions • Testing invariants using assertion 			<p>LO 3. Apply string processing, exceptions, and assertions</p> <p>3.1 Use search, parse, and build strings in accordance with Java framework</p> <p>3.2 Use string formatting in accordance with Java framework</p> <p>3.3 Use Throw and Throws statements in accordance with Java framework</p> <p>3.4 Use Try statement with multi-catch in accordance with Java framework</p> <p>3.5 Use Autoclose features with a try-with-resources statement in accordance with Java framework</p> <p>3.6 Create custom exceptions in accordance with Java framework</p> <p>3.7 Test invariants by using assertions in accordance with Java framework</p>	<p>TLE_ ICTJAVA1 1-12CFJA-IIf- i-39</p>

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
<ul style="list-style-type: none"> • Defining and creating the layout of JDBC API • Using JDBC driver to connect to database • Applying JDBC Row Set Provider, Row Set Factory, and Row Set interfaces • Creating and using Prepared Statement and Callable Statement objects 			<p>LO 4. Building database applications with JBDC</p> <p>4.1 Define layout of the JDBC API in accordance with Java framework</p> <p>4.2 Use JDBC driver to connect to database in accordance with Java framework</p> <p>4.3 Use JDBC Row Set Provider, Row Set Factory, and Row Set interfaces in accordance with Java framework</p> <p>4.4 Create and use Prepared Statement and Callable Statement objects</p>	<p>TLE_ICTJAVA1 1-12CFJA-IIj- IIIa-c-40</p>
<ul style="list-style-type: none"> • CPU Usage monitoring procedures • Network I/O functions and monitoring procedures • Disk I/O functions and monitoring procedures • Network I/O functions and monitoring procedures • Virtual Memory Usage and its monitoring • Java Virtual Machine and its monitoring • Garbage Collection Tuning 			<p>LO 5. Monitor operating system performance</p> <p>5.1 Demonstrate monitoring CPU Usage in accordance with Java framework</p> <p>5.2 Demonstrate monitoring Network I/O in accordance with Java framework</p> <p>5.3 Demonstrate monitoring Disk I/O in accordance with Java framework</p> <p>5.4 Demonstrate monitoring Virtual Memory Usage in accordance with Java framework</p> <p>5.5 Demonstrate monitoring Java Virtual Machine in accordance with Java framework</p> <p>5.6 Demonstrate garbage Collection Tuning in accordance with Java framework</p>	<p>TLE_ICTJAVA1 1-12CFJA- IIIId-g-41</p>
<ul style="list-style-type: none"> • Types of Garbage Collectors • Garbage Collection Algorithms • Creating Garbage Collection Algorithms • JVM Ergonomics • Tuning the garbage collection • Selecting the correct garbage collector • Interpreting the garbage 			<p>LO 6. Work with language-level concerns and garbage collection</p> <p>6.1 Create garbage Collection Algorithms in accordance with Java framework</p> <p>6.2 Demonstrate types of Garbage Collectors in accordance with Java framework</p> <p>6.3 Demonstrate JVM Ergonomics in accordance with Java framework</p> <p>6.4 Tune Garbage Collection in accordance</p>	<p>TLE_ICTJAVA1 1-12CFJA- IIIh-j-IVa-42</p>

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CONTENT	CONTENT STANDARD	PERFORMANCE STANDARD	LEARNING COMPETENCIES	CODE
collection output			with Java framework 6.5 Select correct Garbage Collector in accordance with Java framework 6.6 Interpret Garbage Collection Output in accordance with Java framework	
<ul style="list-style-type: none"> • Use and application of Reference Types • The use of Finalizers • String-efficient Java and its applications • Use and application of Collection Classes • Use and application of Threads • Profile the performance of a Java Application and tune the performance of a Java application at the language level 			LO 7. Work with performance tuning at the language level 7.1 Demonstrate Reference Types in Java in accordance with Java framework 7.2 Demonstrate the use of Finalizers in accordance with Java framework 7.3 Demonstrate string-efficient Java Applications in accordance with Java framework 7.4 Demonstrate Collection Classes in accordance with Java framework 7.5 Use threads in accordance with Java framework 7.6 Use I/O is efficiently in accordance with Java framework	TLE_ICTJAVA1 1-12CFJA-IVb- e-43

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

RESOURCES			METHODOLOGY	ASSESSMENT METHOD
TOOLS	EQUIPMENT	MATERIALS		
<ul style="list-style-type: none"> • Computer Software <ul style="list-style-type: none"> - IDE - Libraries • Internet Access • Application Servers <ul style="list-style-type: none"> - Database - Web 	<ul style="list-style-type: none"> • Network Computer with peripherals • Server • Printer Whiteboard • LCD Projector and Screen • Ergonomic computer tables and chairs 	<ul style="list-style-type: none"> • Learning materials/ guide • Practice materials • Handouts • Reference books 	<ul style="list-style-type: none"> • School-based • Dual Training System • Apprenticeship • Industry Immersion 	<ul style="list-style-type: none"> • Observation in workplace • Demonstration • Oral questioning

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

GLOSSARY

GENERAL

1. Certification - the process of verifying and validating the competencies of a person through assessment.
2. Certificate of Competency (COC) - a certification issued to individuals who pass the assessment for a single unit or cluster of units of competency.
3. Common Competencies - the skills and knowledge needed by all people working in a particular industry.
4. Competency - the possession and application of knowledge, skills, and attitudes to perform work activities to the standard expected in the workplace.
5. Competency Assessment - the process of collecting evidence and making judgments on whether competency has been achieved.
6. Competency Standard (CS) - the industry-determined specification of competencies required for effective work performance.
7. Context of Assessment - refers to the place where assessment is to be conducted or carried out.
8. Core Competencies - the specific skills and knowledge needed in a particular area of work, i.e., industry sector/occupation/job role.
9. Elective Competencies - the additional skills and knowledge required by the individual or enterprise for work.
10. Elements - the building blocks of a unit of competency; they describe in outcome terms the functions that a person must perform in the workplace.
11. Evidence Guide - a component of the unit of competency that defines or identifies the evidence required to determine the competence of the individual. It provides information on critical aspects of competency, underpinning knowledge, underpinning skills, resource implications, assessment method, and context of assessment.
12. Level - refers to the category of skills and knowledge required to do a job.
13. Method of Assessment - refers to the ways of collecting evidence and when evidence should be collected.
14. National Certificate (NC) - a certification issued to individuals who achieve all the required units of competency for a national qualification defined under the Training Regulations. NCs are aligned to specific levels within the PTQF.
15. Performance Criteria - evaluative statements that specify what is to be assessed and the required level of performance.
16. Qualification - a cluster of units of competencies that meets job roles and is significant in the workplace. It is also a certification awarded to a person on successful completion of a course in recognition of having demonstrated competencies in an industry sector.
17. Range of Variables - describes the circumstances or context in which the work is to be performed.
18. Recognition of Prior Learning (RPL) - the acknowledgement of an individual's skills, knowledge, and attitudes gained from life and work experiences outside registered training programs.
19. Resource Implications - refers to the resources needed for the successful performance of the work activity described in the unit of competency. It includes work environment and conditions, materials, tools, and equipment.
20. Basic Competencies - the skills and knowledge that everyone needs for work.
21. Training Regulations (TR) - refers to the document promulgated and issued by TESDA consisting of competency standards, national qualifications and training guidelines for specific sectors/occupations. The TR serves as basis for establishment of qualification and certification under the PTQF. It also serves as guide for development of competency-based curricula and instructional materials including registration of TVET programs offered by TVET providers.
22. Underpinning Knowledge - refers to the competency involved in applying knowledge to perform work activities. It includes specific knowledge that is essential to the performance of the competency.

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

- 23. Underpinning Skills - refers to the list of the skills needed to achieve the elements and performance criteria in the unit of competency. It includes both generic and industry-specific skills.
- 24. Unit of Competency - a component of the competency standards stating a specific key function or role in a particular job or occupation; it is the smallest component of achievement that can be assessed and certified under the PTQF.

SECTOR SPECIFIC

- 25. ADO.NET - a set of computer software components that programmers can use to access data and data services. It is part of the base class library included in the Microsoft .NET Framework. It is commonly used by programmers to access and modify data stored in relational database systems, though it can also access data in nonrelational sources. ADO.NET is sometimes considered an evolution of ActiveX Data Objects (ADO) technology, but was changed so extensively that it can be considered an entirely new product.
- 26. ASP.NET - a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites, web applications and web services. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language.
- 27. Algorithm - a type of effective method in which a list of well-defined instructions for completing a task will, when given an initial state, proceed through a well-defined series of successive states, eventually terminating in an end-state. The transition from one state to the next is not necessarily deterministic; some algorithms, known as probabilistic algorithms, incorporate randomness.
- 28. Artificial intelligence programmer - develops the logic the game uses to carry out a large number of actions. An AI programmer may program pathfinding, strategy, and enemy tactic systems. This is one of the most challenging aspects of game programming and its sophistication is developing rapidly.
- 29. Browser - a software package that provides the user interface for accessing Internet, intranet, and extranet Web sites.
- 30. COBOL - one of the oldest programming languages. Its name is an acronym for COmmon Business-Oriented Language, defining its primary domain in business, finance, and administrative systems for companies and governments.
- 31. Compiler - a computer program (or set of programs) that transforms source code written in a programming language (the source language) into another computer language (the target language, often having a binary form known as object code). The most common reason for wanting to transform source code is to create an executable program.
- 32. Computer - a device that has the ability to accept data, internally store and execute a program of instructions, perform mathematical, logical, and manipulative operations on data, and report the results.
- 33. Computer program - also a software program, or just a program, is a sequence of instructions written to perform a specified task for a computer. A computer requires programs to function, typically executing the program's instructions in a central processor. The program has an executable form that the computer can use directly to execute the instructions. The same program in its human-readable source code form, from which executable programs are derived (e.g., compiled), enables a programmer to study and develop its algorithms.
- 34. Computer programming - the iterative process of writing or editing source code. Editing source code involves testing, analyzing, and refining, and sometimes coordinating with other programmers on a jointly developed program. A person who practices this skill is referred to as a computer programmer, software developer, or coder. The sometimes lengthy process of computer programming is usually referred to as software development.
- 35. Computer Terminal - any input/output device connected by telecommunications links to a computer.
- 36. C language - an imperative (procedural) systems implementation language. It was designed to be compiled using a relatively straightforward compiler, to provide low-level access to memory, to provide language constructs that map efficiently to machine instructions, and to require minimal run-time support. C was therefore useful for many applications that had formerly been coded in assembly language.

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

- 37. C++ language - a statically typed, free-form, multi-paradigm, compiled, general-purpose programming language. It comprises a combination of both high-level and low-level language features. Some of its application domains include systems software, application software, device drivers, embedded software, high-performance server and client applications, and entertainment software such as video games.
- 38. Data - objective measurements of the attributes (characteristics) of entities such as people, places, things, and events.
- 39. Data access - typically refers to software and activities related to storing, retrieving, or acting on data housed in a database or other repository. Historically, different methods and languages were required for every repository, including each different database, file system, etc., and many of these repositories stored their content in different and incompatible formats.
- 40. Decompiler - the name given to a computer program that performs the reverse operation to that of a compiler. That is, it translates a file containing information at a relatively low level of abstraction (usually designed to be computer readable rather than human readable) into a form having a higher level of abstraction (usually designed to be human readable).
- 41. Documentation - a collection of documents or information.
- 42. Edit - to modify the form or format of data.
- 43. End-user - anyone who uses an information system or the information it produces.
- 44. Ergonomics - the science and technology emphasizing the safety, comfort, and ease of use of human-operated machines. The goal of ergonomics is to produce systems that are user-friendly, safe, comfortable, and easy to use.
- 45. HTML - HyperText Markup Language, the predominant markup language for Web pages. It is written in the form of HTML elements consisting of "tags" surrounded by angle brackets within the Web page content. It is the building blocks of all basic websites.
- 46. Information - data placed in a meaningful and useful context for an end user.
- 47. Information and Communication Technology (ICT) - refers to technologies associated with the transmission and exchange of data in the form of sound, text, visual images, signals, or any combination of those forms through the use of digital technology. It encompasses such services as telecommunications, posts, multimedia, electronic commerce, broadcasting, and information technology.
- 48. Integrated development environment (IDE) - a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, a compiler and/or interpreter, build automation tools, and (usually) a debugger. Typically, an IDE is dedicated to a specific programming language, so as to provide a feature set which most closely matches the programming paradigms of the language. However, some multiple-language IDEs are in use, such as Eclipse, ActiveState Komodo, recent versions of NetBeans, and Microsoft Visual Studio.
- 49. Java - a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". Java is currently one of the most popular programming languages in use, and is widely used from application software to web applications
- 50. Java applications - typically compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture.
- 51. Local Area Network (LAN) - a communications network that typically connects computers, terminals, and other computerized devices within a limited physical area such as an office, building, manufacturing, plant and other worksites.
- 52. Microsoft .NET Framework - a software framework that can be installed on computers running Microsoft Windows operating systems. It includes a large library of coded solutions to common programming problems and a common language infrastructure that manages the execution of programs written specifically for the framework. The .NET Framework supports multiple programming languages in a manner that allows language interoperability, whereby each language can utilize code written in other languages; in particular, the .NET library is available to all the programming languages that .NET encompasses.
- 53. Microsoft SQL Server - a relational model database server produced by Microsoft. Its primary query languages are T-SQL and ANSI SQL.

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

54. Object code - or an object file; the representation of code that a compiler or assembler generates by processing a source code file. Object files contain compact code, often called "binaries". A linker is typically used to generate an executable or library by linking object files together. The only essential element in an object file is machine code (code directly executed by a computer's CPU). Object files for embedded systems might contain nothing but machine code. However, object files often also contain data for use by the code at runtime, relocation information, program symbols (names of variables and functions) for linking and/or debugging purposes, and other debugging information.
55. Oracle - the Oracle Database (commonly referred to as Oracle RDBMS or simply as Oracle) is an object-relational database management system [2] produced and marketed by Oracle Corporation.
56. Oracle Forms - a software product for creating screens that interact with an Oracle database. It has a typical IDE including an object navigator, property sheet, and code editor that uses PL/SQL. It was originally developed to run server-side in character mode terminal sessions. It was ported to other platforms, including Windows, to function in a client–server environment. Later versions were ported to Java where it runs in a Java EE container and can integrate with Java and Web services. The primary focus of Forms is to create data entry systems that access an Oracle database.
57. Oracle Reports - a tool for developing reports against data stored in an Oracle database. Oracle Reports consists of Oracle Reports Developer (a component of the Oracle Developer Suite) and Oracle Application Server Reports Services (a component of the Oracle Application Server).
58. Outsourcing - turning over all or part of an organization’s information systems operation to outside contractors, known as systems integrators or facilities management companies.
59. Programming language - an artificial language designed to express computations that can be performed by a machine, particularly a computer. Programming languages can be used to create programs that control the behavior of a machine, to express algorithms precisely, or as a mode of human communication.
60. Quality Assurance - methods for ensuring that information systems are free from errors and fraud and provide information products of high quality.
61. Relational database management system (RDBMS) - a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd. Most popular commercial and open source databases currently in use are based on the relational database model. A short definition of an RDBMS may be a DBMS in which data is stored in the form of tables and the relationship among the data is also stored in the form of tables.
62. Service-oriented programming (SOP) - a programming paradigm that uses "services" as the unit of computer work, to design and implement integrated business applications and mission-critical software programs. Services can represent steps of business processes and thus one of the main applications of this paradigm is the cost-effective delivery of standalone or composite business applications that can “integrate from the inside-out.”
63. Software - computer programs and procedures concerned with the operation of an information system.
64. Source code - any collection of statements or declarations written in some human-readable computer programming language. Source code is the means most often used by programmers to specify the actions to be performed by a computer.
65. SQL - often referred to as Structured Query Language; a database computer language designed for managing data in relational database management systems (RDBMS), and originally based upon relational algebra. Its scope includes data insert, query, update and delete, schema creation and modification, and data access control.
66. Standards - measures of performance developed to evaluate the progress of a system toward its objectives.
67. System - an assembly of methods, procedures, or techniques unified by regulated interaction to form an organized whole.
68. User- friendly - a characteristic of human-operated equipment and systems that makes them safe, comfortable, and easy to use.

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

- 69. User interface
 - the system by which people (users) interact with a machine. The user interface includes hardware (physical) and software (logical) components. User interfaces exist for various systems, and provide a means of: 1) input, allowing the users to manipulate a system, and/or 2) output, allowing the system to indicate the effects of the users' manipulation.
- 70. VB.NET
 - a redesigned, object-oriented dialect of Visual Basic.
- 71. Web application
 - an application that is accessed over a network such as the Internet or an intranet. The term may also mean a computer software application that is hosted in a browser-controlled environment (e.g., a Java applet) or coded in a browser-supported language (such as JavaScript, combined with a browser-rendered markup language like HTML) and reliant on a common Web browser to render the application executable.
- 72. Windows Forms
 - the name given to the graphical application programming interface (API) included as a part of Microsoft's .NET Framework, providing access to the native Microsoft Windows interface elements by wrapping the existing Windows API in managed code. While it is seen as a replacement for the earlier and more complex C++ based Microsoft Foundation Class Library, it does not offer a paradigm comparable to model–view–controller.
- 73. Windows Forms application
 - an event-driven application supported by Microsoft's .NET Framework. Unlike a batch program, it spends most of its time simply waiting for the user to do something, such as fill in a text box or click a button.

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

CODE BOOK LEGEND

Sample: **TLE_**ICT**JAVA**11-12**POAD**-I**f**-i-29

LEGEND		SAMPLE		DOMAIN / COMPONENT	CODE
First Entry	Learning Area and Strand/ Subject or Specialization	Technology and Livelihood Education	TLE_	Basic Competencies	
		Information and Communications Technology	ICT	Lead Workplace Communication	LWC
		Programming (JAVA) NC III	JAVA	Lead Small Teams	LST
	Grade Level	Grade 11 to 12	11-12	Develop and Practice Negotiation Skills	DPN
Uppercase Letter/s	Domain/Content/Component/Topic	Performing Object-Oriented Analysis and Design	POAD	Solve Problems Related to Work Activities	PRW
				Use Mathematical Concepts and Techniques	MCT
				Use Relevant Technologies	URT
				Common Competencies	
Roman Numeral *Zero if no specific Quarter	Quarter	Second Quarter	II	Apply Quality Standards	AQS
				Perform Computer Operations	PCO
Lower case letter/s *put a hyphen (-) in between letters to indicate more than a specific week	Week	Week six to nine	f-i	Core Competencies	
				Perform Object-Oriented Analysis and Design	POAD
				Create And Fine-Tune Java Technology Application Using Object-Oriented Program Concepts	CFJA
Arabic Number	Learning Competency	Apply basics of Java language	29		

Technology-Livelihood Education and Technical-Vocational Track specializations may be taken between Grades 9 to 12.

Schools may offer specializations from the four strands as long as the minimum number of hours for each specialization is met.

Please refer to the sample Curriculum Map on the next page for the number of semesters per ICT specialization and those that have pre-requisites. Curriculum Maps may be modified according to specializations offered by a school.

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
 (320 hours)

SAMPLE ICT CURRICULUM MAP (as of May 2016)**

Grade 7/8 (EXPLORATORY)	GRADES 9-12			
EXPLORATORY	Computer Systems Servicing (NC II)⁺ updated based on TESDA Training Regulations released December 28, 2007 8 sems			
			*Telecom OSP and Subscriber Line Installation (Copper Cable/POTS and DSL) (NC II) 4 sems	
			*Telecom OSP Installation (Fiber Optic Cable) (NC II) 2 sems	*Broadband Installation (Fixed Wireless Systems) (NC II) 2 sems
	Illustration (NC II) 4 sems	Technical Drafting (NC II) 4 sems		
	Computer Programming (.Net Technology) (NC III)⁺ updated based on TESDA Training Regulations released December 28, 2013 4 sems	Contact Center Services (NC II) 4 sems		
	Computer Programming (Java) (NC III)⁺ updated based on TESDA Training Regulations released December 28, 2013 4 sems	Animation (NC II) 4 sems		
	Computer Programming (Oracle Database) (NC III)⁺ updated based on TESDA Training Regulations released December 28, 2013 4 sems	Medical Transcription (NC II) 4 sems		

* Please note that these subjects have pre-requisites mentioned in the CG.
 + CG updated based on new Training Regulations of TESDA.
■ Pre-requisites of the subjects to the right should be taken up during these semesters.

****This is just a sample. Schools make their own curriculum maps considering the specializations to be offered. Subjects may be taken up at any point during Grades 9-12.**

K to 12 BASIC EDUCATION CURRICULUM
JUNIOR HIGH SCHOOL TECHNICAL LIVELIHOOD EDUCATION AND SENIOR HIGH SCHOOL - TECHNICAL-VOCATIONAL-LIVELIHOOD TRACK
INFORMATION AND COMMUNICATIONS TECHNOLOGY–PROGRAMMING (JAVA) NC III
(320 hours)

Reference:

Technical Education and Skills Development Authority-Qualification Standards Office. *Training Regulations for Programming (Java) NC III*. Taguig City, Philippines: TESDA, 2013.