

TRAINING REGULATIONS



PROGRAMMING (JAVA) NC III

**INFORMATION AND COMMUNICATION
TECHNOLOGY (ICT) SECTOR**

TECHNICAL EDUCATION AND SKILLS DEVELOPMENT AUTHORITY
East Service Road, South Superhighway, Taguig City, Metro Manila

TABLE OF CONTENTS

ICT SECTOR

Programming (Java) - NATIONAL CERTIFICATE LEVEL III

	Page No.	
SECTION 1	■ DESCRIPTION OF PROGRAMMING (JAVA) NC III QUALIFICATION	1-3
SECTION 2	■ COMPETENCY STANDARDS	4 - 39
	• Basic Competencies	4 – 22
	• Common Competencies	23 - 29
	• Core Competencies	30 - 39
SECTION 3	■ TRAINING STANDARDS	40 - 45
	3.1 Curriculum Design	40 - 42
	3.2 Training Delivery	43
	3.3 Trainee Entry Requirements	44
	3.4 List of Tools, Equipment and Materials	44
	3.5 Training Facilities	45
	3.6 Trainers Qualification	45
	3.7 Institutional Assessment	45
SECTION 4	■ NATIONAL ASSESSMENT AND CERTIFICATION ARRANGEMENTS	46
COMPETENCY MAP		47
DEFINITION OF TERMS		48 – 53
ACKNOWLEDGEMENT		54

TRAINING REGULATIONS FOR PROGRAMMING (JAVA) NC III

Section 1 PROGRAMMING (JAVA) NC III QUALIFICATIONS

This TESDA Course is delivered in cooperation with Oracle's Workforce Development Program (WDP). Oracle Corporation launched WDP to address the continued global information technology (IT) worker shortage and need for accessible and low cost IT skills training. WDP enables participating educational institutions to deliver Oracle training in full-time and part-time programs for students in their local communities. WDP training will prepare students for Oracle certification testing. With the widely-acclaimed credential of Oracle certification, WDP students are readied for a variety of entry-level Oracle job-roles as well as career advancement.

The **Programming (Java) NC III** Qualification consists of competencies that a person must achieve to develop or write program codes using a personal computer or workstation as part of a systems development team. It includes core competencies on programming language on Java, such as to develop command-line/console and desktop applications and to develop enterprise/web applications using Java technology.

At the end of the course, participants should be equipped with sufficient knowledge to take and pass the internationally acknowledged Oracle Java SE Certified Associate and Certified Professional Exam and participant will be awarded with an Oracle Certified Professional Java Programmer.

This Qualification is packaged from the competency map of the Information and Communication Technology Industry (Service sector) as shown in Annex A.

Benefits of Taking this course and Oracle Certification

- Oracle's Java Programmer certification is critical to maximizing your potential and progressing your Java career.
- Over 30, 000 Java Developers per year become certified on Java.
- More than 800,000 of these Java developers are using the knowledge, skills and recognition learned through Java Certifications to improve their job prospects, earn more money* and become more valuable employees.

With 3 billion devices running Java worldwide, the world's top companies rely on experienced Java Programmers to build and maintain these critical applications.

- Java is the number one development platform.
- 97% of desktops run Java.
- Java is the number one language for enterprise development.

Java Programmers Are In Demand

Because of the prevalence of Java, there is a continued demand for well-trained, highly-skilled Java programmers to create and maintain critical applications.

Becoming Java Certified

The process of becoming Java certified broadens your knowledge and deepens your skills through exposure to a wide array of important features, functions and tasks.

As you prepare for these certification exams, you'll work your way through hands-on exercises and labs using real Java code, which expands your capabilities as a Java developer. Additionally, the certification exams validate your capabilities using real-world, scenario-based questions that assess and challenge your ability to think and perform.

Accelerate Your Development Career

Ultimately, becoming Oracle Certified makes you a better Java developer. You'll become more valuable to your current and/or future potential employer, increase job security and improve your future prospects.

Get started today by reviewing the Java Certification levels and requirements below.

** According to Oracle's 2012 Salary and Satisfaction Survey, 100% of Programmers and Developers surveyed reported receiving a promotion, compensation increase or other improvements to their careers as a result of becoming certified.*

Oracle Certification Explanations:



The Oracle Certified Associate (OCA) credential is typically the first step toward achieving the flagship Oracle Certified Professional certification. The OCA credential ensures that the individual is equipped with fundamental skills, providing a strong foundation for supporting Oracle products. An OCA credential is available for several of today's most in-demand technology job roles.



The Oracle Certified Professional (OCP) credential is the benchmark of professional skill and technical expertise required to manage, develop, or implement enterprise-wide databases, middleware, or applications. Increasingly, IT managers use the OCP credential to evaluate the qualifications of employees and job candidates.

The units of competency comprising this qualification include the following:

CODE NO.	BASIC COMPETENCIES
5 00 311 1 09	Lead workplace communication
5 00 311 1 10	Lead small teams
5 00 311 1 11	Develop and practice negotiation skills
5 00 311 1 12	Solve problems related to work activities
5 00 311 1 13	Use mathematical concepts and techniques
5 00 311 1 14	Use relevant technologies

CODE NO.	COMMON COMPETENCIES
ICT315202	Apply quality standards
ICT311203	Perform computer operations

CODE NO.	CORE COMPETENCIES
ICT313359	Perform object-oriented analysis and design in Java technology
ICT313360	Create and fine-tune Java technology applications using object-oriented programming concept

A person who has achieved this Qualification can be employed in any or more of the following:

- Java Programmer
- Java Programming support staff
- Software developer
- Application developer
- User Interface Developer

SECTION 2 COMPETENCY STANDARDS

This section gives the details of the contents of the basic, common and core units of competency required in **Programming (Java) NC III**.

BASIC COMPETENCIES

UNIT OF COMPETENCY: LEAD WORKPLACE COMMUNICATION

UNIT CODE : **500311109**

UNIT DESCRIPTOR : This unit covers the knowledge, skills and attitudes required to lead in the dissemination and discussion of ideas, information and issues in the workplace.

ELEMENT	PERFORMANCE CRITERIA <i>Italicized</i> terms are elaborated in the Range of Variables
1. Communicate information about workplace processes	1.1. Appropriate <i>communication method</i> is selected 1.2. Multiple operations involving several topics areas are communicated accordingly 1.3. Questions are used to gain extra information 1.4. Correct sources of information are identified 1.5. Information is selected and organized correctly 1.6. Verbal and written reporting is undertaken when required 1.7. Communication skills are maintained in all situations
2. Lead workplace discussions	2.1. Response to workplace issues are sought 2.2. Response to workplace issues are provided immediately 2.3. Constructive contributions are made to workplace discussions on such issues as production, quality and safety 2.4. Goals/objectives and action plan undertaken in the workplace are communicated
3. Identify and communicate issues arising in the workplace	3.1. Issues and problems are identified as they arise 3.2. Information regarding problems and issues are organized coherently to ensure clear and effective communication 3.3. Dialogue is initiated with appropriate personnel 3.4. Communication problems and issues are raised as they arise

RANGE OF VARIABLES

VARIABLE	RANGE
1. Methods of communication	1.1. Non-verbal gestures 1.2. Verbal 1.3. Face to face 1.4. Two-way radio 1.5. Speaking to groups 1.6. Using telephone 1.7. Written 1.8. Internet

EVIDENCE GUIDE

<p>1. Critical aspects of Competency</p>	<p>Assessment requires evidence that the candidate:</p> <ul style="list-style-type: none"> 1.1. Dealt with a range of communication/information at one time 1.2. Made constructive contributions in workplace issues 1.3. Sought workplace issues effectively 1.4. Responded to workplace issues promptly 1.5. Presented information clearly and effectively written form 1.6. Used appropriate sources of information 1.7. Asked appropriate questions 1.8. Provided accurate information
<p>2. Underpinning knowledge</p>	<ul style="list-style-type: none"> 2.1. Organization requirements for written and electronic communication methods 2.2. Effective verbal communication methods
<p>3. Underpinning Skills</p>	<ul style="list-style-type: none"> 3.1. Organize information 3.2. Understand and convey intended meaning 3.3. Participate in variety of workplace discussions 3.4. Comply with organization requirements for the use of written and electronic communication methods
<p>4. Resource Implications</p>	<p>The following resources MUST be provided:</p> <ul style="list-style-type: none"> 4.1. Variety of Information 4.2. Communication tools 4.3. Simulated workplace
<p>5. Methods of Assessment</p>	<p>Competency may be assessed through:</p> <ul style="list-style-type: none"> 5.1. Competency in this unit must be assessed through 5.2. Direct Observation 5.3. Interview
<p>6. Context for Assessment</p>	<ul style="list-style-type: none"> 6.1. Competency may be assessed in the workplace or in simulated workplace environment

UNIT OF COMPETENCY: LEAD SMALL TEAMS**UNIT CODE : 500311110****UNIT DESCRIPTOR :** This unit covers the knowledge, skills and attitudes to lead small teams including setting and maintaining team and individual performance standards.

ELEMENT	PERFORMANCE CRITERIA <i>Italicized</i> terms are elaborated in the Range of Variables
1. Provide team leadership	1.1. Work requirements are identified and presented to team members 1.2. Reasons for instructions and requirements are communicated to team members 1.3. Team members' queries and concerns are recognized, discussed and dealt with
2. Assign responsibilities	2.1. Duties, and responsibilities are allocated having regard to the skills, knowledge and aptitude required to properly undertake the assigned task and according to company policy 2.2. Duties are allocated having regard to individual preference, domestic and personal considerations, whenever possible
3. Set performance expectations for team members	3.1. Performance expectations are established based on client needs and according to assignment requirements 3.2. Performance expectations are based on individual team members duties and area of responsibility 3.3. Performance expectations are discussed and disseminated to individual team members
4. Supervise team performance	4.1. Monitoring of performance takes place against defined performance criteria and/or assignment instructions and corrective action taken if required 4.2. Team members are provided with feedback , positive support and advice on strategies to overcome any deficiencies 4.3. Performance issues which cannot be rectified or addressed within the team are referenced to appropriate personnel according to employer policy 4.4. Team members are kept informed of any changes in the priority allocated to assignments or tasks which might impact on client/customer needs and satisfaction 4.5. Team operations are monitored to ensure that employer/client needs and requirements are met 4.6. Follow-up communication is provided on all issues affecting the team 4.7. All relevant documentation is completed in accordance with company procedures

RANGE OF VARIABLES

VARIABLE	RANGE
1. Work requirements	1.1. Client Profile 1.2. Assignment instructions
2. Team member's concerns	2.1. Roster/shift details
3. Monitor performance	3.1. Formal process 3.2. Informal process
4. Feedback	4.1. Formal process 4.2. Informal process
5. Performance issues	5.1. Work output 5.2. Work quality 5.3. Team participation 5.4. Compliance with workplace protocols 5.5. Safety 5.6. Customer service

EVIDENCE GUIDE

<p>1. Critical Aspects of Competency</p>	<p>Assessment requires evidence that the candidate:</p> <ol style="list-style-type: none"> 1.1. Maintained or improved individuals and/or team performance given a variety of possible scenario 1.2. Assessed and monitored team and individual performance against set criteria 1.3. Represented concerns of a team and individual to next level of management or appropriate specialist and to negotiate on their behalf 1.4. Allocated duties and responsibilities, having regard to individual's knowledge, skills and aptitude and the needs of the tasks to be performed 1.5. Set and communicated performance expectations for a range of tasks and duties within the team and provided feedback to team members
<p>2. Underpinning Knowledge</p>	<ol style="list-style-type: none"> 2.1. Company policies and procedures 2.2. Relevant legal requirements 2.3. How performance expectations are set 2.4. Methods of Monitoring Performance 2.5. Client expectations 2.6. Team member's duties and responsibilities
<p>3. Underpinning Skills</p>	<ol style="list-style-type: none"> 3.1. Communication skills required for leading teams 3.2. Informal performance counseling skills 3.3. Team building skills 3.4. Negotiating skills
<p>4. Resource Implications</p>	<p>The following resources MUST be provided:</p> <ol style="list-style-type: none"> 4.1. Access to relevant workplace or appropriately simulated environment where assessment can take place 4.2. Materials relevant to the proposed activity or task
<p>5. Methods of Assessment</p>	<p>Competency may be assessed through:</p> <ol style="list-style-type: none"> 5.1. Direct observations of work activities of the individual member in relation to the work activities of the group 5.2. Observation of simulation and/or role play involving the participation of individual member to the attainment of organizational goal 5.3. Case studies and scenarios as a basis for discussion of issues and strategies in teamwork
<p>6. Context of Assessment</p>	<ol style="list-style-type: none"> 6.1. Competency assessment may occur in workplace or any appropriately simulated environment 6.2. Assessment shall be observed while task are being undertaken whether individually or in-group

UNIT OF COMPETENCY: DEVELOP AND PRACTICE NEGOTIATION SKILLS

UNIT CODE : **500311111**

UNIT DESCRIPTOR : This unit covers the skills, knowledge and attitudes required to collect information in order to negotiate to a desired outcome and participate in the negotiation.

ELEMENT	PERFORMANCE CRITERIA <i>Italicized</i> terms are elaborated in the Range of Variables
1. Plan negotiations	1.1 Information on <i>preparing for negotiation</i> is identified and included in the plan 1.2 Information on creating <i>non verbal environments</i> for positive negotiating is identified and included in the plan 1.3 Information on <i>active listening</i> is identified and included in the plan 1.4 Information on different <i>questioning techniques</i> is identified and included in the plan 1.5 Information is checked to ensure it is correct and up-to- date
2. Participate in negotiations	2.1 Criteria for successful outcome are agreed upon by all parties 2.2 Desired outcome of all parties are considered 2.3 Appropriate language is used throughout the negotiation 2.4 A variety of questioning techniques are used 2.5 The issues and processes are documented and agreed upon by all parties 2.6 Possible solutions are discussed and their viability assessed 2.7 Areas for agreement are confirmed and recorded 2.8 Follow-up action is agreed upon by all parties

RANGE OF VARIABLES

VARIABLE	RANGE
1. Preparing for negotiation	1.1 Background information on other parties to the negotiation 1.2 Good understanding of topic to be negotiated 1.3 Clear understanding of desired outcome/s 1.4 Personal attributes 1.4.1 self awareness 1.4.2 self esteem 1.4.3 objectivity 1.4.4 empathy 1.4.5 respect for others 1.5 Interpersonal skills 1.5.1 listening/reflecting 1.5.2 non verbal communication 1.5.3 assertiveness 1.5.4 behavior labeling 1.5.5 testing understanding 1.5.6 seeking information 1.5.7 self disclosing 1.6 Analytic skills 1.6.1 observing differences between content and process 1.6.2 identifying bargaining information 1.6.3 applying strategies to manage process 1.6.4 applying steps in negotiating process 1.6.5 strategies to manage conflict 1.6.6 steps in negotiating process 1.6.7 options within organization and externally for resolving conflict
2. Non-verbal environments	2.1 Friendly reception 2.2 Warm and welcoming room 2.3 Refreshments offered 2.4 Lead in conversation before negotiation begins
3. Active listening	3.1 Attentive 3.2 Don't interrupt 3.3 Good posture 3.4 Maintain eye contact 3.5 Reflective listening
4. Questioning techniques	4.1 Direct 4.2 Indirect 4.3 Open-ended

EVIDENCE GUIDE

<p>1. Critical Aspects of Competency</p>	<p>Assessment requires evidence that the candidate:</p> <ul style="list-style-type: none"> 1.1 Demonstrated sufficient knowledge of the factors influencing negotiation to achieve agreed outcome 1.2 Participated in negotiation with at least one person to achieve an agreed outcome
<p>2. Underpinning Knowledge and Attitude</p>	<ul style="list-style-type: none"> 2.1 Codes of practice and guidelines for the organization 2.2 Organizations policy and procedures for negotiations 2.3 Decision making and conflict resolution strategies procedures 2.4 Problem solving strategies on how to deal with unexpected questions and attitudes during negotiation 2.5 Flexibility 2.6 Empathy
<p>3. Underpinning Skills</p>	<ul style="list-style-type: none"> 3.1 Interpersonal skills to develop rapport with other parties 3.2 Communication skills (verbal and listening) 3.3 Observation skills 3.1 Negotiation skills
<p>4. Resource Implications</p>	<p>The following resources MUST be provided:</p> <ul style="list-style-type: none"> 4.1 Room with facilities necessary for the negotiation process 4.2 Human resources (negotiators)
<p>5. Methods of Assessment</p>	<p>Competency may be assessed through:</p> <ul style="list-style-type: none"> 5.1 Observation/demonstration and questioning 5.2 Portfolio assessment 5.3 Oral and written questioning 5.4 Third party report
<p>6. Context for Assessment</p>	<ul style="list-style-type: none"> 6.1 Competency to be assessed in real work environment or in a simulated workplace setting.

UNIT OF COMPETENCY: SOLVE PROBLEMS RELATED TO WORK ACTIVITIES

UNIT CODE : 500311112

UNIT DESCRIPTOR : This unit of covers the knowledge, skills and attitudes required to solve problems in the workplace including the application of problem solving techniques and to determine and resolve the root cause of problems.

ELEMENT	PERFORMANCE CRITERIA <i>Italicized</i> terms are elaborated in the Range of Variables
1. Identify the problem	1.1. Variances are identified from normal operating parameters; and product quality 1.2. Extent, cause and nature are of the problem are defined through observation, investigation and analytical techniques 1.3. Problems are clearly stated and specified
2. Determine fundamental causes of the problem	2.1. Possible causes are identified based on experience and the use of problem solving tools / analytical techniques. 2.2. Possible cause statements are developed based on findings 2.3. Fundamental causes are identified per results of investigation conducted
3. Determine corrective action	3.1. All possible options are considered for resolution of the problem 3.2. Strengths and weaknesses of possible options are considered 3.3. Corrective actions are determined to resolve the problem and possible future causes 3.4. Action plans are developed identifying measurable objectives, resource needs and timelines in accordance with safety and operating procedures
4. Provide recommendation/s to manager	4.1. Report on recommendations are prepared 4.2. Recommendations are presented to appropriate personnel. 4.3. Recommendations are followed-up, if required

RANGE OF VARIABLES

VARIABLE	RANGE
1. Analytical techniques	1.1. Brainstorming 1.2. Intuitions/Logic 1.3. Cause and effect diagrams 1.4. Pareto analysis 1.5. SWOT analysis 1.6. Gant chart, Pert CPM and graphs 1.7. Scatter grams
2. Problem	2.1. Non – routine process and quality problems 2.2. Equipment selection, availability and failure 2.3. Teamwork and work allocation problem 2.4. Safety and emergency situations and incidents
3. Action plans	3.1. Priority requirements 3.2. Measurable objectives 3.3. Resource requirements 3.4. Timelines 3.5. Co-ordination and feedback requirements 3.6. Safety requirements 3.7. Risk assessment 3.8. Environmental requirements

EVIDENCE GUIDE

<p>1. Critical Aspects of Competency</p>	<p>Assessment requires evidence that the candidate:</p> <ol style="list-style-type: none"> 1.1. Identified the problem 1.2. Determined the fundamental causes of the problem 1.3. Determined the correct / preventive action 1.4. Provided recommendation to manager <p>These aspects may be best assessed using a range of scenarios / case studies / what ifs as a stimulus with a walk through forming part of the response. These assessment activities should include a range of problems, including new, unusual and improbable situations that may have happened.</p>
<p>2. Underpinning Knowledge</p>	<ol style="list-style-type: none"> 2.1. Competence includes a thorough knowledge and understanding of the process, normal operating parameters, and product quality to recognize non-standard situations 2.2. Competence to include the ability to apply and explain, sufficient for the identification of fundamental cause, determining the corrective action and provision of recommendations <ol style="list-style-type: none"> 2.2.1. Relevant equipment and operational processes 2.2.2. Enterprise goals, targets and measures 2.2.3. Enterprise quality, OHS and environmental requirement 2.2.4. Principles of decision making strategies and techniques 2.2.5. Enterprise information systems and data collation 2.2.6. Industry codes and standards
<p>3. Underpinning Skills</p>	<ol style="list-style-type: none"> 3.1. Using range of formal problem solving techniques 3.2. Identifying and clarifying the nature of the problem 3.3. Devising the best solution 3.4. Evaluating the solution 3.5. Implementation of a developed plan to rectify the problem

4. Resource Implications	4.1. Assessment will require access to an operating plant over an extended period of time, or a suitable method of gathering evidence of operating ability over a range of situations. A bank of scenarios / case studies / what ifs will be required as well as bank of questions which will be used to probe the reason behind the observable action.
5. Methods of Assessment	<p>Competency may be assessed through:</p> <p>5.1. Case studies on solving problems in the workplace</p> <p>5.2. Observation</p> <p>The unit will be assessed in a holistic manner as is practical and may be integrated with the assessment of other relevant units of competency. Assessment will occur over a range of situations, which will include disruptions to normal, smooth operation. Simulation may be required to allow for timely assessment of parts of this unit of competency. Simulation should be based on the actual workplace and will include walk through of the relevant competency components.</p>
6. Context of Assessment	6.1. In all workplace, it may be appropriate to assess this unit concurrently with relevant teamwork or operation units.

UNIT OF COMPETENCY: USE MATHEMATICAL CONCEPTS AND TECHNIQUES

UNIT CODE : 500311113

UNIT DESCRIPTOR : This unit covers the knowledge, skills and attitudes required in the application of mathematical concepts and techniques.

ELEMENT	PERFORMANCE CRITERIA <i>Italicized</i> terms are elaborated in the Range of Variables
1. Identify mathematical tools and techniques to solve problem	1.1 Problem areas are identified based on given condition 1.2 <i>Mathematical techniques</i> are selected based on the given problem
2. Apply mathematical procedure/solution	2.1 Mathematical techniques are applied based on the problem identified 2.2 Mathematical computations are performed to the level of accuracy required for the problem 2.3 Results of mathematical computation is determined and verified based on job requirements
3. Analyze results	3.1 Result of application is reviewed based on expected and required specifications and outcome 3.2 <i>Appropriate action</i> is applied in case of error

RANGE OF VARIABLES

VARIABLE	RANGE
1. Mathematical techniques	May include but are not limited to: 1.1 Four fundamental operations 1.2 Measurements 1.3 Use/Conversion of units of measurements 1.4 Use of standard formulas
2. Appropriate action	2.1 Review in the use of mathematical techniques (e.g. recalculation, re-modeling) 2.2 Report error to immediate superior for proper action

EVIDENCE GUIDE

1. Critical Aspects of Competency	Assessment requires evidence that the candidate: 1.1 Identified, applied and reviewed the use of mathematical concepts and techniques to workplace problems
2. Underpinning Knowledge	2.1 Fundamental operation (addition, subtraction, division, multiplication) 2.2 Measurement system 2.3 Precision and accuracy 2.4 Basic measuring tools/devices
3. Underpinning Skills	3.1 Applying mathematical computations 3.2 Using calculator 3.3 Using different measuring tools
4. Resource Implications	The following resources MUST be provided: 4.1 Calculator 4.2 Basic measuring tools 4.3 Case Problems
5. Methods of Assessment	Competency may be assessed through: 5.1 Authenticated portfolio 5.2 Written Test 5.3 Interview/Oral Questioning 5.4 Demonstration
6. Context of Assessment	6.1 Competency may be assessed in the work place or in a simulated work place setting

UNIT OF COMPETENCY: USE RELEVANT TECHNOLOGIES

UNIT CODE : 500311114

UNIT DESCRIPTOR : This unit of competency covers the knowledge, skills, and attitude required in selecting, sourcing and applying appropriate and affordable technologies in the workplace.

ELEMENT	PERFORMANCE CRITERIA <i>Italicized</i> terms are elaborated in the Range of Variables
1. Study/select appropriate technology	1.1 Usage of different technologies is determined based on job requirements 1.2 Appropriate technology is selected as per work specification
2. Apply relevant technology	2.1 Relevant technology is effectively used in carrying out function 2.2 Applicable software and hardware are used as per task requirement 2.3 Management concepts are observed and practiced as per established industry practices
3. Maintain/enhance relevant technology	3.1 Maintenance of technology is applied in accordance with the industry standard operating procedure, manufacturer's operating guidelines and occupational health and safety procedure to ensure its operative ability 3.2 Updating of technology is maintained through continuing education or training in accordance with job requirement 3.3 Technology failure/ defect is immediately reported to the concern/responsible person or section for appropriate action

RANGE OF VARIABLES

VARIABLE	RANGE
1. Technology	May include but are not limited to: 1.1 Office technology 1.2 Industrial technology 1.3 System technology 1.4 Information technology 1.5 Training technology
2. Management concepts	May include but not limited to: 2.1 Real Time Management 2.2 KAIZEN or continuous improvement 2.3 5s 2.4 Total Quality Management 2.5 Other management/productivity tools
3. Industry standard operating procedure	3.1 Written guidelines relative to the usage of office technology/equipment 3.2 Verbal advise/instruction from the co-worker
4. Manufacturer's operating guidelines/ instructions	4.1 Written instruction/manuals of specific technology/ equipment 4.2 General instruction manual 4.3 Verbal advise from manufacturer relative to the operation of equipment
5. Occupational health and safety procedure	5.1 Relevant statutes on OHS 5.2 Company guidelines in using technology/equipment
6. Appropriate action	6.1 Implementing preventive maintenance schedule 6.2 Coordinating with manufacturer's technician

EVIDENCE GUIDE

1. Critical Aspects of Competency	Assessment requires evidence that the candidate: 1.1 Studied and selected appropriate technology consistent with work requirements 1.2 Applied relevant technology 1.3 Maintained and enhanced operative ability of relevant technology
2. Underpinning Knowledge	2.1 Awareness on technology and its function 2.2 Repair and maintenance procedure 2.3 Operating instructions 2.4 Applicable software 2.5 Communication techniques 2.6 Health and safety procedure 2.7 Company policy in relation to relevant technology 2.8 Different management concepts 2.9 Technology adaptability
3. Underpinning Skills	3.1 Relevant technology application/implementation 3.2 Basic communication skills 3.3 Software applications skills 3.4 Basic troubleshooting skills
4. Resource Implications	The following resources MUST be provided: 4.1 Relevant technology 4.2 Interview and demonstration questionnaires 4.3 Assessment packages
5. Methods of Assessment	Competency must be assessed through: 5.1 Interview 5.2 Actual demonstration 5.3 Authenticated portfolio (related certificates of training/seminar)
6. Context of Assessment	6.1 Competency may be assessed in actual workplace or simulated environment

COMMON COMPETENCIES

UNIT TITLE : **APPLY QUALITY STANDARDS**

UNIT CODE : **506315202**

UNIT DESCRIPTOR : This unit covers the knowledge, skills, attitudes and values needed to apply quality standards in the workplace. The unit also includes the application of relevant safety procedures and regulations, organization procedures and customer requirements.

ELEMENT	PERFORMANCE CRITERIA
1. Assess quality of received materials	<p>1.1. Work instruction is obtained and work is carried out in accordance with standard operating procedures.</p> <p>1.2. Received materials are checked against workplace standards and specifications.</p> <p>1.3. Faulty materials related to work are identified and isolated.</p> <p>1.4. Faults and any identified causes are recorded and/or reported to the supervisor concerned in accordance with workplace procedures.</p> <p>1.5. Faulty materials are replaced in accordance with workplace procedures.</p>
2. Assess own work	<p>2.1. Documentation relative to quality within the company is identified and used.</p> <p>2.2. Completed work is checked against workplace standards relevant to the task undertaken.</p> <p>2.3. Errors are identified and isolated.</p> <p>2.4. Information on the quality and other indicators of production performance are recorded in accordance with workplace procedures.</p> <p>2.5. In cases of deviations from specific quality standards, causes are documented and reported in accordance with the workplace's standards operating procedures.</p>
3. Engage in quality improvement	<p>3.1. Process improvement procedures are participated in relative to workplace assignment.</p> <p>3.2. Work is carried out in accordance with process improvement procedures.</p> <p>3.3. Performance of operation or quality of product of service to ensure customer satisfaction is monitored.</p>

RANGE OF VARIABLES

VARIABLE	RANGE
1 Materials	1.1 Materials may include but not limited to: 1.1.1. Manuals 1.1.2. Job orders 1.1.3. Instructional videos
2 Faults	2.1 Faults may include but not limited to: 2.1.1. Materials not to specification 2.1.2. Materials contain incorrect/outdated information 2.1.3. Hardware defects 2.1.4. Materials that do not conform with any regulatory agencies
3 Documentation	3.1 Organization work procedures 3.2 Manufacturer's instruction manual 3.3 Customer requirements 3.4 Forms
4 Errors	4.1 Errors may be related but not limited to the following: 4.1.1. Deviation from the requirements of the Client 4.1.2. Deviation from the requirement of the organization
5 Quality standards	5.1 Quality standards may be related but not limited to the following: 5.1.1. Materials 5.1.2. Hardware 5.1.3. Final product 5.1.4. Production processes 5.1.5. Customer service
6 Customer	6.1 Co-worker 6.2 Supplier/Vendor 6.3 Client 6.4 Organization receiving the product or service

EVIDENCE GUIDE

<p>1 Critical aspect of competency</p>	<p>Assessment must show that the candidate:</p> <ul style="list-style-type: none"> 1.1 Carried out work in accordance with the company's standard operating procedures 1.2 Performed task according to specifications 1.3 Reported defects detected in accordance with standard operating procedures 1.4 Carried out work in accordance with the process improvement procedures
<p>2 Underpinning knowledge</p>	<ul style="list-style-type: none"> 2.1 Relevant production processes, materials and products 2.2 Characteristics of materials, software and hardware used in production processes 2.3 Quality checking procedures 2.4 Workplace procedures 2.5 Safety and environmental aspects of production processes 2.6 Fault identification and reporting 2.7 Quality improvement processes
<p>3 Underpinning skills</p>	<ul style="list-style-type: none"> 3.1 Reading skills required to interpret work instruction 3.2 Communication skills needed to interpret and apply defined work procedures 3.3 Carry out work in accordance with OHS policies and procedures 3.4 Critical thinking 3.5 Solution providing and decision-making
<p>4 Method of assessment</p>	<p>The assessor must select two of the following to objectively evaluate the candidate:</p> <ul style="list-style-type: none"> 4.1 Observation and oral questioning 4.2 Third party report 4.3 Portfolio 4.4 Practical demonstration
<p>5 Resource implication</p>	<p>Materials, software and hardware to be used in a real or simulated situation</p>
<p>6 Context of Assessment</p>	<p>Assessment may be conducted in the workplace or in a simulated environment</p>

UNIT TITLE : PERFORM COMPUTER OPERATIONS

UNIT CODE : 506311203

UNIT DESCRIPTOR : This unit covers the knowledge, skills, attitudes and values needed to perform computer operations which include inputting, accessing, producing and transferring data using the appropriate hardware and software.

ELEMENT	PERFORMANCE CRITERIA <i>Italicized</i> terms are elaborated in the Range of Variables
1. Plan and prepare for task to be taken undertaken	1.1. Requirements of task are determined in accordance with the required output. 1.2. Appropriate hardware and software are selected according to task assigned and required outcome. 1.3. Task is planned to ensure that OH & S guidelines and procedures are followed. 1.4. Client -specific guidelines and procedures are followed. 1.5. Required data security guidelines are applied in accordance with existing procedures.
2. Input data into computer	2.1. Data are entered into the computer using appropriate program/application in accordance with company procedures. 2.2. Accuracy of information is checked and information is saved in accordance with standard operating procedures. 2.3. Inputted data is stored in storage media according to requirements. 2.4. Work is performed within ergonomic guidelines .
3. Access information using computer	3.1. Correct program/application is selected based on job requirements. 3.2. Program/application containing the information required is accessed according to company procedures. 3.3. Desktop icons are correctly selected, opened and closed for navigation purposes. 3.4. Keyboard techniques are carried out in line with OH & S requirements for safe use of keyboards.

<p>4. Produce output/ data using computer system</p>	<p>4.1. Entered data are processed using appropriate software commands.</p> <p>4.2. Data are printed out as required using computer hardware /peripheral devices in accordance with standard operating procedures.</p> <p>4.3. Files and data are transferred between compatible systems using computer software, hardware/peripheral devices in accordance with standard operating procedures.</p>
<p>5. Use basic functions of a www-browser to locate information</p>	<p>5.1. Information requirements for internet search are established.</p> <p>5.2. Browser is launched.</p> <p>5.3. Search engine is loaded.</p> <p>5.4. Appropriate search criteria/or URL of site is entered.</p> <p>5.5. Relevant links are followed to locate required information.</p> <p>5.6. Useful pages are bookmarked or printed as required.</p>
<p>6. Maintain computer equipment and systems</p>	<p>6.1. Procedures for ensuring security of data, including regular back-ups and virus checks are implemented in accordance with standard operating procedures.</p> <p>6.2. Basic file maintenance procedures are implemented in line with the standards operating procedures.</p>

RANGE OF VARIABLES

VARIABLE	RANGE
1 Hardware and peripheral devices	1.1 Personal computers 1.2 Networked systems 1.3 Communication equipment 1.4 Printers 1.5 Scanners 1.6 Keyboard 1.7 Mouse 1.8 Voice/Data logger
2 Software	Software includes the following but not limited to: 2.1 Word processing packages 2.2 Database packages 2.3 Internet 2.4 Spreadsheets 2.5 Client Specific Software
3 OH & S guidelines	3.1 OHS guidelines 3.2 Enterprise procedures
4 Storage media	Storage media include the following but not limited to: 4.1 Diskettes 4.2 CDs 4.3 Zip disks 4.4 hard disk drives, local and remote 4.5 Optical drives
5 Ergonomic guidelines	5.1 Types of equipment used 5.2 Appropriate furniture 5.3 Seating posture 5.4 Lifting posture 5.5 Visual display unit screen brightness
6 Desktop icons	6.1 Icons include the following but not limited to: 6.2 Directories/folders 6.3 Files 6.4 Network devices 6.5 Recycle bin 6.6 Program icons
7 Maintenance	7.1 Creating and managing more space in the hard disk and other peripherals 7.2 Reviewing programs 7.3 Deleting unwanted files 7.4 Backing up files 7.5 Checking hard drive for errors 7.6 Using up to date anti-virus programs 7.7 Cleaning dust from internal and external surfaces

EVIDENCE GUIDE

<p>1 Critical aspects of competency</p>	<p>Assessment must show that the candidate:</p> <ul style="list-style-type: none"> 1.1 Selected and used hardware components correctly and according to the task requirement 1.2 used basic software applications to create new files and documents 1.3 Produced accurate and complete data in accordance with the requirements 1.4 Used appropriate devices and procedures to transfer files/data accurately 1.5 Used basic functions of a www-browser to locate information.
<p>2 Underpinning knowledge</p>	<ul style="list-style-type: none"> 2.1 Basic ergonomics of keyboard and computer user 2.2 Main types of computers and basic features of different operating systems 2.3 Main parts of a computer 2.4 Storage devices and basic categories of memory 2.5 Relevant types of software 2.6 General security, privacy legislation and copyright 2.7 Viruses 2.8 OH & S principles and responsibilities 2.9 Calculating computer capacity 2.10 Productivity Application 2.11 Business Application 2.12 System Software
<p>3 Underpinning skills</p>	<ul style="list-style-type: none"> 3.1 Reading and comprehension skills required to interpret work instruction and to interpret basic user manuals. 3.2 Communication skills to identify lines of communication, request advice, follow instructions and receive feedback. 3.3 Technology skills to use equipment safely including keyboard skills.
<p>4 Method of assessment</p>	<p>The assessor may select two of the following assessment methods to objectively assess the candidate:</p> <ul style="list-style-type: none"> 4.1 Direct Observation and Oral Questioning 4.2 Practical demonstration
<p>5 Resource implication</p>	<ul style="list-style-type: none"> 5.1 Computer hardware with peripherals 5.2 Appropriate software
<p>6 Context of Assessment</p>	<ul style="list-style-type: none"> 6.1 Assessment may be conducted in the workplace or in a simulated environment

CORE COMPETENCIES

UNIT OF COMPETENCY: PERFORM OBJECT-ORIENTED ANALYSIS AND DESIGN IN JAVA TECHNOLOGY

UNIT CODE: ICT313359

UNIT DESCRIPTOR: This unit covers the knowledge, skills and attitude needed to use various Java programming language constructs to create several Java technology applications. Specifically, it consist of competencies in applying basics of Java language, working with methods and encapsulation, and working with inheritance and handling exceptions.

It also focuses on effectively using object-oriented technologies and the use of software modeling, as applied to a software development process. Students will learn to present one practical & complete object-oriented analysis and design (OOAD) roadmap, and will get hands-on experience, from requirements gathering to system design.

ELEMENT	PERFORMANCE CRITERIA <i>Italicized</i> terms are elaborated in the Range of Variables
1 Apply Basics of Java language	1.1 Executable Java applications are created in accordance with Java framework 1.2 Java packages are imported to make them accessible in the code 1.3 Working with Java Data types is demonstrated in accordance with Java framework 1.4 Using Operators and Decision Constructs is demonstrated in accordance with Java framework 1.5 Creating and Using Arrays is demonstrated in accordance with Java framework 1.6 Using Loop Constructs is demonstrated in accordance with Java framework
2 Work with Methods and Encapsulation	2.1 Methods with arguments and return values are created in accordance with Java framework 2.2 Static keywords are applied to methods and fields in accordance with Java framework 2.3 Overloaded method is created in accordance with Java framework 2.4 Access modifiers are applied in accordance with Java framework 2.5 Encapsulation principled are applied to a class in accordance with Java framework

<p>3 Work with Inheritance and Handling Exceptions</p>	<p>3.1 Inheritance is implemented in accordance with Java framework 3.2 Code that demonstrates the use of polymorphism is developed in accordance with Java framework 3.3 Super and this syntax are used to access objects and constructors in accordance with Java framework 3.4 Abstract classes and interfaces are used in accordance with Java framework 3.5 How exceptions alter normal program flow are determined by creating a try-catch block.</p>
<p>4. Examine Object-Oriented Concepts and Terminology</p>	<p>4.1 Important object-oriented (OO) concepts are described in accordance with Java framework 4.2 Fundamental OO terminology are defined in accordance with Java framework</p>
<p>5. Explain Modeling and Software Development Process</p>	<p>5.1 Object-Oriented Software Development (OOSD) process is explained in accordance with Java framework 5.2 Benefits of modeling software are explained in accordance with Java framework 5.3 Purpose, activities, and artifacts of the following OOSD workflows (disciplines) are explained</p>
<p>6. Create Use Case Diagrams and Use Case Scenarios</p>	<p>6.1 The need for a Use Case Diagram is justified in accordance with Java framework 6.2 Use Case Diagram for a software system is developed based on the goals of the business owner 6.3 Use Case Diagrams is developed based on the goals of all the stakeholders 6.4 Use Case form is created describing a summary of the scenarios in the main and alternate flows</p>
<p>7. Transition Analysis to Design using Interaction Diagrams</p>	<p>7.1 Purpose and elements of the Design model are explained in accordance with Java framework 7.2 Essential elements of a UML Communication diagram are identified in accordance with Java framework 7.3 Communication diagram view of the Design model is created in accordance with Java framework 7.4 Sequence diagram view of the Design model is created in accordance with Java framework</p>
<p>8. Introduce Architectural Concepts and Architecture Tiers Diagrams</p>	<p>8.1 Difference between architecture and design is distinguished in accordance with Java framework 8.2 Tiers, layers, and systemic qualities are described in accordance with Java framework 8.3 Architecture workflow is described in accordance with Java framework 8.4 Architecture Tiers are defined in accordance with Java framework</p>

RANGE OF VARIABLES:

VARIABLE	RANGE
1. Executable Java applications	<ul style="list-style-type: none"> • Hello World • Hello with name • Hello with name and date
2. Working with Java Data Types	<ul style="list-style-type: none"> • Declare and initialize variables • Differentiate between object references and primitive variables • Read and write to object fields • Explain an object's lifecycle (creation, dereference, and garbage collection) • Call methods on objects • Manipulate data using StringBuilder class and its methods • Create and manipulate Strings
3. Using Operators and Decision Constructs	<ul style="list-style-type: none"> • Use Java operators • Use parenthesis to override operator precedence • Test equality between strings and other objects using == and equals() • Create and use if-else constructs • Use a switch statement
4. Creating and Using Arrays	<ul style="list-style-type: none"> • Declare, initialize, and use a one-dimensional array • Declare, initialize, and use a multi-dimensional array • Declare and use an ArrayList
5. Using Loop Constructs	<ul style="list-style-type: none"> • Create and use while loops • Create and use for loops including the enhanced for loop • Create and use do-while loops • Compare loop constructs • Use break and continue
6. OOSD Workflows	<ul style="list-style-type: none"> • Requirements Gathering • Requirements Analysis, • Architecture and Design, Implementation • Testing & Deployment
7. Defining Architecture Tiers	<ul style="list-style-type: none"> • Describe the concepts of the Client and Presentation tiers • Describe the concepts of the Business tier • Describe the concepts of the Resource and Integration tiers • Describe the concepts of the Solution model

EVIDENCE GUIDE

<p>1. Critical aspect of competency</p>	<p>Assessment requires evidence that the candidate:</p> <ol style="list-style-type: none"> 1.1 Applied Basics of Java language 1.2 Worked with Methods and Encapsulation 1.3 Worked with Inheritance and Handling Exceptions 1.4 Use object-oriented technologies and perform object-oriented analysis and design. 1.5 Follow a software development process using an OO software project. 1.6 Use the widely adopted graphical modeling language - the Unified Modeling Language (UML) version 2.2. 1.7 Manage complexity of artifacts; describe the problem and proposed solution.
<p>2. Underpinning Knowledge</p>	<ol style="list-style-type: none"> 2.1. 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 2.2. Uses of decision and looping constructs and methods to dictate program flow 2.3. Basic error handling for your Java technology programs 2.4. 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 2.5. Java technology and the Java programming language 2.6. Understanding of basic object oriented concepts such as inheritance, encapsulation, and abstraction 2.7. Uses and manipulation of object references to write simple error handling code <ul style="list-style-type: none"> ○ Differentiate among checked exceptions, RuntimeExceptions, and Error ○ Describe what exceptions are used for in Java ○ Invoke a method that throws an Exception ○ Recognize common exception classes and categories 2.8 Recognize and document use case dependencies using UML notation for extends, includes, and generalization 2.9 Describe how to manage the complexity of Use Case Diagrams by creating UML packaged views Identify and document scenarios for a use case 2.10 Describe how to reference included and extending use cases 2.11 Identify and document non-functional requirements (NFRs), business rules, risks, and priorities for a use case 2.12 Identify the purpose of a Supplementary Specification Document 2.13 Identify the essential elements in an Activity diagram

3. Underpinning skills	3.1. Basic computer operation skills 3.2. Logic analysis 3.3. Communication skills 3.4. Code writing and debugging skills
4. Method of assessment	4.1 The assessor will assess candidate with- <ul style="list-style-type: none"> • Portfolio <ul style="list-style-type: none"> ○ Result of Oracle Vendor Examinations ○ Sample codes • Interview
5.Resource implication	7.1 Computer with: <ul style="list-style-type: none"> 7.1.1. UML tools 7.1.2. Integrated Development Environment for Java 7.2 Access to internet 7.3 Conducive testing environment
6.Context of assessment	Assessment may take place in an accredited Oracle testing center

UNIT OF COMPETENCY: CREATE AND FINE TUNE JAVA TECHNOLOGY APPLICATIONS USING OBJECT-ORIENTED PROGRAMMING CONCEPT

UNIT CODE : ICT313360

UNIT DESCRIPTION : This unit covers the core Application Programming Interfaces (API) used to design object-oriented applications with Java. Students will learn how to write database programs with JDBC. This unit also covers performance-tuning methodologies, performance tuning theories and practical tips on solving difficult performance problems for Java applications. It also covers how to implement interfaces and handle Java programming exceptions, while using object-oriented programming techniques.

ELEMENT	PERFORMANCE CRITERIA <i>Italicized</i> terms are elaborated in the Range of Variables
1. Apply Basics of Java Class Design	1.1 Access modifiers are used in accordance with Java framework 1.2 Instance of operator and casting are used in accordance with Java framework 1.3 Virtual method invocation is used in accordance with Java framework 1.4 Override methods from the Object class are used to improve the functionality of class 1.5 Package and import statements are used in accordance with Java framework
2. Apply Java Advanced Class Design and Object Oriented Design Principles	2.1 Proper use of when and how to apply abstract classes is identified 2.2 Abstract Java classes and subclasses are constructed in accordance with Java framework 2.3 Static and final keywords are used in accordance with Java framework 2.4 Top-level and nested classes are created in accordance with Java framework 2.5 Enumerated types are used in accordance with Java framework 2.6 Write code that declares, implements, and/or extends interfaces 2.7 Object Oriented Design Principles are applied in accordance with Java framework 2.8 Generics and Collections Principles are used in accordance with Java framework
3. Apply String Processing, Exceptions and Assertions	3.1 Search, parse, and build strings are used in accordance with Java framework 3.2 String formatting is used in accordance with Java framework 3.3 Throw and throws statements are used in accordance with Java framework 3.4 Try statement with multi-catch is used in accordance with Java framework

	<p>3.5 Autoclose features with a try-with-resources statement are used in accordance with Java framework</p> <p>3.6 Custom exceptions are created in accordance with Java framework</p> <p>3.7 Invariants by using assertions are tested in accordance with Java framework</p>
4. Build Database Applications with JDBC	<p>4.1 Layout of the JDBC API is defined in accordance with Java framework</p> <p>4.2 JDBC driver is used to connect to database in accordance with Java framework</p> <p>4.3 JDBC RowSetProvider, RowSetFactory, and RowSet interfaces are used in accordance with Java framework</p> <p>4.4 PreparedStatement and CallableStatement objects are created and used</p>
5. Monitor Operating System Performance	<p>5.1 Monitoring CPU Usage is demonstrated in accordance with Java framework</p> <p>5.2 Monitoring Network I/O is demonstrated in accordance with Java framework</p> <p>5.3 Monitoring Disk I/O is demonstrated in accordance with Java framework</p> <p>5.4 Monitoring Virtual Memory Usage is Demonstrated in accordance with Java framework</p> <p>5.5 Monitoring Java Virtual Machine is demonstrated in accordance with Java framework</p> <p>5.6 Garbage Collection Tuning is demonstrated in accordance with Java framework</p>
6. Work with Language Level Concerns and Garbage Collection	<p>6.1 Garbage Collection Algorithms are created in accordance with Java framework</p> <p>6.2 Types of Garbage Collectors are demonstrated in accordance with Java framework</p> <p>6.3 JVM Ergonomics are demonstrated in accordance with Java framework</p> <p>6.4 Garbage Collection is tuned in accordance with Java framework</p> <p>6.5 Correct Garbage Collector is selected in accordance with Java framework</p> <p>6.6 Garbage Collection Output is interpreted in accordance with Java framework</p>
7. Work with Performance Tuning at the Language Level	<p>7.1 Reference Types in Java are demonstrated in accordance with Java framework</p> <p>7.2 The use of Finalizers is demonstrated in accordance with Java framework</p> <p>7.3 String-efficient Java Applications are demonstrated in accordance with Java framework</p> <p>7.4 Collection Classes are demonstrated in accordance with Java framework</p> <p>7.5 Threads are used in accordance with Java framework</p> <p>7.6 I/O is efficiently used in accordance with Java framework</p>

RANGE OF VARIABLES

VARIABLE	RANGE
1. Access Modifiers	<ul style="list-style-type: none"> • private, • protected, • public
2. Object Class	<ul style="list-style-type: none"> • (hashCode(), • equals(), • toString())
3. Object Oriented Design Principles	<ul style="list-style-type: none"> • Choose between interface inheritance and class inheritance • Develop code that implements IS-A and/or HAS-A relationships (apply high cohesion and low coupling principles) • Apply object composition principles (including HAS-A relationships) • Design a class using the Singleton design pattern • Write code to implement the Data Access Object (DAO) pattern • Design and create objects using a Factory, and use factories from the API
4. Generics and Collections Principles	<ul style="list-style-type: none"> • Use the diamond syntax to create a collection and for type inference • Analyze the interoperability of collections that use raw types and generic types • Use wrapper classes, autoboxing and unboxing • Create and use List, Set, and Deque implementations • Create and use a Map • Use java.util.Comparator and java.lang.Comparable • Sort and search arrays and lists
5. Build Strings	<ul style="list-style-type: none"> • Scanner • StringTokenizer • StringBuilder • String • Formatter
6. Java Virtual Machine	<ul style="list-style-type: none"> • HotSpot Generational Garbage Collector • Monitor the Garbage Collector with Command Line Tools • Monitor the Garbage Collector with VisualVM • Monitor the JIT Compiler • Throughput and Responsiveness
7. Garbage Collection Tuning	<ul style="list-style-type: none"> • Find Memory Leaks • Identify Lock Contention • Heap Profiling Anti-patterns • Method Profiling Anti-patterns • Garbage Collection • Generational Garbage Collection • GC Performance Metrics

EVIDENCE GUIDE

<p>1. Critical Aspect of Competency</p>	<p>1.1 Performed multiple operations on database tables, including creating, reading, updating and deleting using JDBC technology</p> <p>1.2 Process strings using a variety of regular expressions and create high-performing multi-threaded applications that avoid deadlock</p> <p>1.3 Implement input/output (I/O) functionality to read from and write to data and text files and understand advanced I/O streams</p> <p>1.4 Create Java technology applications that leverage the object-oriented features of the Java language, such as encapsulation, inheritance, and polymorphism</p> <p>1.5 Set up a performance-tuning environment</p> <p>1.6 Monitor Java applications</p> <p>1.7 Apply rigor to the task of performance tuning</p> <p>1.8 Use various tools and mechanisms for monitoring, profiling and tuning Java applications</p>
<p>2. Underpinning Knowledge</p>	<p>2.1 Knowledge of Java I/O Fundamentals</p> <ul style="list-style-type: none"> • Read and write data from the console • Use streams to read and write files <p>2.2 Knowledge Java File I/O (NIO.2)</p> <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 DirectoryStream and FileVisitor interfaces • Find a class using the PathMatcher class • Watch a directory for changes by using WatcherService <p>2.3 Understand fundamentals of Java Virtual Machine</p> <ul style="list-style-type: none"> • Performance Principles • Common Performance Problems • Performance Methodology • Development and Performance <p>2.4 Apply basic performance tuning principles to a Java application</p> <p>2.5 Profile the performance of a Java Application and tune the performance of a Java application at the language level</p>
<p>3. Underpinning Skills</p>	<p>3.1 Basic computer operation skills</p> <p>3.2 Logic analysis</p> <p>3.3 Communication skills</p> <p>3.4 Code writing and debugging skills</p>

4. Method of Assessment	<p>The assessor will assess candidate with-</p> <p>4.1 Portfolio</p> <ul style="list-style-type: none"> ○ Result of Oracle Vendor Examinations ○ Sample codes <p>4.2 Interview</p>
5. Resource Implication	<p>5.1 Computer with:</p> <ul style="list-style-type: none"> • UML tools • Integrated Development Environment for Java <p>5.2 Access to internet</p> <p>5.3 Conducive testing environment</p>
6. Context of Assessment	Assessment may take place in an accredited Oracle testing center

SECTION 3 TRAINING STANDARDS

These guidelines are set to provide the Technical and Vocational Education and Training (TVET) providers with information and other important requirements to consider when designing training programs for **Programming (Java) NC III**.

3.1 CURRICULUM DESIGN

Course Title: Java Programming

NC Level: NC III

Nominal Training Duration: 64 hrs – Basic Competencies
16 hrs – Common Competencies
160 hrs – Core Competencies

240 hours

Course Description:

This course is designed to develop & enhance the knowledge, skills, & attitudes of a programmer in accordance with industry standards. It covers the basic, common & core competencies on programming language, specifically Java. The nominal duration of **240** hours covers the required units at Java Programming NC III. TVET providers can however, offer a longer, ladderized course covering the basic, common and core plus specialized competency unit/s.

BASIC COMPETENCIES

64 hrs

Unit of Competency	Learning Outcomes	Methodology	Assessment Approach
1. Lead workplace communication	1.1 Communicate information about workplace processes. 1.2 Lead workplace discussions. 1.3 Identify and communicate issues arising in the workplace	<ul style="list-style-type: none"> • Group discussion • Role Play • Brainstorming 	<ul style="list-style-type: none"> • Observation • Interviews
2. Lead small teams	2.1 Provide team leadership. 2.2 Assign responsibilities among members. 2.3 Set performance expectation for team members. 2.4 Supervise team performance	<ul style="list-style-type: none"> • Lecture • Demonstration • Self-paced (modular) 	<ul style="list-style-type: none"> • Demonstration • Case studies

3. Develop and practice negotiation skills	3.1 Identify relevant information in planning negotiations 3.2 Participate in negotiations 3.3 Document areas for agreement	<ul style="list-style-type: none"> • Direct observation • Simulation/role playing • Case studies 	<ul style="list-style-type: none"> • Written test • Practical/ performance test
4. Solve workplace problem related to work activities	4.1 Explain the analytical techniques. 4.2 Identify the problem. 4.3 Determine the possible cause/s of the problem.	<ul style="list-style-type: none"> • Direct observation • Simulation/role playing • Case studies 	<ul style="list-style-type: none"> • Written test • Practical/ performance test
5. Use mathematical concepts and techniques	5.1 Identify mathematical tools and techniques to solve problem 5.2 Apply mathematical procedures/solution 5.3 Analyze results	<ul style="list-style-type: none"> • Direct observation • Simulation/role playing • Case studies 	<ul style="list-style-type: none"> • Written test • Practical/ performance test
6. Use relevant technologies	6.1 Identify appropriate technology 6.2 Apply relevant technology 6.3 Maintain/enhance relevant technology	<ul style="list-style-type: none"> • Direct observation • Simulation/role playing • Case studies 	<ul style="list-style-type: none"> • Written test • Practical/ performance test

COMMON COMPETENCIES

16 hrs.

Unit of Competency	Learning Outcomes	Methodology	Assessment Approach
1. Apply Quality Standards	1.1 Asses quality of received materials 1.2 Assess own work 1.3 Engage in quality improvement	<ul style="list-style-type: none"> ▪ Field trip ▪ Symposium ▪ Film showing ▪ Simulation ▪ On the job training 	<ul style="list-style-type: none"> ▪ Demonstration & questioning ▪ Observation & questioning ▪ Third party report
2. Operate a Personal Computer	2.1 Plan and prepare for task to be undertaken 2.2 Input data into computer 2.3 Access information using computer 2.4 Produce output/data using computer system 2.5 Use basic functions of a web browser to locate information 2.6 Maintain computer equipment and systems	<ul style="list-style-type: none"> • Modular • Film showing • Computer based training (e-learning) • Project method • On the job training 	<ul style="list-style-type: none"> • Demonstration & questioning • Observation & questioning • Third party report • Assessment of output product • Portfolio • Computer- based assessment

CORE COMPETENCIES

160 hrs.

Unit of Competency	Learning Outcome	Methodology	Assessment Approach
<p>1. Perform object-oriented analysis and design in Java technology</p>	<p>1.1 Apply basics of Java language 1.2 Work with methods and encapsulation 1.3 Work with inheritance and handling exceptions. 1.4 Examine object-oriented concepts and terminology 1.5 Explain modeling and software development process 1.6 Create Use Case Diagrams and Use Case Scenarios 1.7 Transition analysis to design using Interaction Diagrams 1.8 Introduce Architectural Concepts and Architecture Tiers Diagrams</p>	<ul style="list-style-type: none"> • Lecture/ Discussion • Hands on • Exercises • Demonstration 	<ul style="list-style-type: none"> • Practical exam • Interviews/ questioning
<p>2. Create and fine tune Java technology applications using object-oriented programming concept</p>	<p>2.1 Apply Basics of Java Class Design 2.2 Apply Java Advance Class Design and Object Oriented Design Principles 2.3 Apply string processing, exceptions and assertions 2.4 Build database applications with JDBC 2.5 Monitor operating system performance 2.6 Work with language level concerns and garbage collection 2.7 Work with performance tuning at the language level</p>	<ul style="list-style-type: none"> • Lecture/ Discussion • Hands on • Exercises • Demonstration 	<ul style="list-style-type: none"> • Practical exam • Interviews/ questioning

3.1. TRAINING DELIVERY

The delivery of training should adhere to the design of the curriculum. Delivery should be guided by the 10 basic principles of the competency-based TVET.

- The training is based on curriculum developed from the competency standards;
- Learning is modular in its structure;
- Training delivery is individualized and self-paced;
- Training is based on work that must be performed;
- Training materials are directly related to the competency standards and the curriculum modules;
- Assessment is based in the collection of evidence of the performance of work to the industry required standard;
- Training is based both on and off-the-job components;
- Allows for recognition of prior learning (RPL) or current competencies;
- Training allows for multiple entry and exit; and
- Approved training programs are nationally accredited.

The competency-based TVET system recognizes various types of delivery modes, both on and off-the-job as long as the learning is driven by the competency standards specified by the industry. The following training modalities may be adopted when designing training programs:

- The dualized mode of training delivery is preferred and recommended. Thus programs would contain both in-school and in-industry training or fieldwork components. Details can be referred to the Dual Training System (DTS) Implementing Rules and Regulations.
- Modular/self-paced learning is a competency-based training modality wherein the trainee is allowed to progress at his own pace. The trainer facilitates the training delivery.
- Peer teaching/mentoring is a training modality wherein fast learners are given the opportunity to assist the slow learners.
- Supervised industry training or on-the-job training is an approach in training designed to enhance the knowledge and skills of the trainee through actual experience in the workplace to acquire a specific competencies prescribed in the training regulations.
- Distance learning is a formal education process in which majority of the instruction occurs when the students and instructors are not in the same place. Distance learning may employ correspondence study, or audio, video or computer technologies.

3.2. TRAINEE ENTRY REQUIREMENTS

Trainees or students should possess the following requirements:

- Must have completed at least 10 yrs. basic education or an ALS grade 10 certificate of rating holder Must have completed at least 10 yrs. basic education or an ALS grade 10 certificate of rating holder
- can communicate either oral or written;
- with basic computer skills

This list does not include specific institutional requirements such as educational attainment, appropriate work experience, and others that may be required of the trainees by the school or training center delivering the TVET program.

3.3. LIST OF TOOLS, EQUIPMENT AND MATERIALS

Recommended list of tools, equipment and materials for the conduct of training in **Programming (Java) NC III**

TOOLS		EQUIPMENT		MATERIALS	
Qty	Description	Qty	Description	Qty	Description
As required	<i>Computer Software</i> e.g. - <i>IDE</i> - <i>Libraries</i>	15	Network Computer with peripherals	As required	Learning materials/ guide
		1	Server		Practice materials
		1	Printer		Hand-outs
		1	White board		Reference books
1	Internet access	1	LCD Projector and screen		
As required	Application servers e.g. - database - web	15	Ergonomic computer tables and chairs		

The quantity of tools and equipment to be used for the conduct of training for this qualification shall depend on the number of students, size of the class, and/or modality of training. The most important consideration is to make sure that tools and equipment are adequately provided to all trainees when needed. The actual list of tools, equipment, machines, supplies and other materials to be used shall be identified and detailed in the Competency Based Curriculum (CBC) to be submitted by the TVET provider when registering a course or training program with TESDA.

Due to the fast-changing nature of the Information and Communications Technology (ICT) sector, TVET providers are reminded to use and provide their trainees with the latest technology tools, equipment and materials where appropriate and applicable.

In cases where there are specialized tools, equipment and facilities that are not generally considered standard requirements or not absolute requisites for training, the industry working group or TESDA may provide guidelines or specific advice on such matters.

3.4. TRAINING FACILITIES

The Programming workshop must be of concrete structure. Based on class size of 15 student/trainees the space requirements for the teaching/learning and circulation areas are as follows:

TEACHING/LEARNING AREAS	SIZE IN METERS	AREA IN SQ. METERS	QTY	TOTAL AREA IN SQ. METERS
Lecture Area	5 x 8	40	1	40
Computer laboratory	6 x 8	48	1	48
Learning Resource Area	4 x 5	20	1	20
Wash ,Toilet & Locker Room	2 x 5	10	2	20
Total				128
Facilities / Equipment / Circulation**				38
Total Area				166

**** Area requirement is equivalent to 30% of the total teaching/learning areas**

Note: The training center must be accredited by Oracle Philippines and must be a member of the Workforce Development Program.

3.5. TRAINERS QUALIFICATION

Programming (Java) NC III

- Must be Oracle Certified Professional Java SE7
- Must be certified and approved by Oracle University Philippines
- Must be a holder of National TVET Trainer's Certificate (NTTC) Level 1
- Must be physically & mentally fit.

3.6. INSTITUTIONAL ASSESSMENT

Institutional Assessment is undertaken by trainees to determine their achievement of units of competency. A certificate of achievement is issued for each unit of competency. The institutional assessment is administered by the trainer/assessor.

The result of the institutional assessment may be considered as an evidence for national assessment.

SECTION 4 NATIONAL ASSESSMENT AND CERTIFICATION ARRANGEMENTS

- 4.1 To attain the National Qualification of **Programming (Java) NC III**, the candidate must demonstrate competency in all the units listed in Section 1. Successful candidates shall be awarded a **National Certificate level III** signed by the TESDA Director General.
- 4.2 The qualification of Programming (Java) NC III may be attained by passing the following exams:
- 4.2.1. Certification Exam 1 - Java SE 7 Programmer I 1Z0-803 which will assess the following unit of competencies:
- Perform object-oriented analysis and design in Java technology
- 4.2.2. Certification Exam 2 – Java SE7 Programmer II – 1Z0 – 804 which will assess the following unit of competencies:
- Create and fine tune Java technology applications using object-oriented programming concept
- Upon accumulation and submission of Certification Exams 1 and 2, an individual shall be issued the corresponding **National Certificate**.
- 4.3 Assessment shall focus on the core units of competency. The basic and common units shall be integrated or assessed concurrently with the core units.
- 4.4 The following are qualified to apply for assessment and certification:
- 4.4.1. Graduate of formal, non-formal, and informal, including enterprise-based, training programs.
- 4.4.2. Experienced workers (wage employed or self-employed)
- 4.5 The guidelines on assessment and certification are discussed in detail in the “Procedures Manual on Assessment and Certification” and “Guidelines on the Implementation of the Philippine TVET Qualification and Certification System (PTQCS)”.

COMPETENCY MAP – INFORMATION AND COMMUNICATIONS TECHNOLOGY (ICT) SECTOR

BASIC COMPETENCIES

Receive and Respond to workplace communication	Work with Others	Demonstrate Work Values	Participate in Workplace Communication	Work in a Team Environment	Practice Career Professionalism	Practice occupational Health and Safety Procedures
Practice Housekeeping Procedures (5S)	Lead Workplace Communication	Lead Small Team	Develop and Practice Negotiation Skills	Solve Problems Related to Work Activities	Use Mathematical Concepts and Techniques	Use Relevant Technologies
Utilize Specialized Communication skills	Develop Team and Individual	Apply Problem Solving Techniques in the Workplace	Collect, Analyze and Organize Information	Plan and Organize Work	Promote Environmental Protection	

COMMON COMPETENCIES

Perform Computer Operation	Apply Quality Standards
-----------------------------------	--------------------------------

Programming (Java) NC III

CORE COMPETENCIES

Communicate Effectively using the English Language	Deliver Quality Customer Service	Communicate Effectively in a Customer Contact Center	Utilize Enterprise/ Company Technology	Conduct Contact Center Campaign	Provide Specialized Support and Assistance to Customers	Manage the Activities of a Contact Center Work Team
Demonstrate Understanding and Knowledge of the American Culture and Geography	Perform Basic Computer Operation and Internet Navigation	Lead a Contact Center Work Team	Use Business Technology	Use Medical Terminology to Carry Out Task	Produce Text from Audio Transcription	Produce Cleaned-up and In-betweened Drawings
Review / Edit Documents	Manage the Activities of a Work Team	Lead a Team in Delivering Quality Service	Produce Over-all Designs for Animation	Create 2D Models and Images	Produce 2D Colored Animation	Produce Key Drawings for Animation
Produce Background Designs	Composit and Edit Animation Sequence	Produce Storyboard for Animation	Create 3D Models and Images	Coordinate the Production of Animation	Use email and search the web using browsers	Install Computer Systems and Networks
Configure Computer Systems and Networks	Diagnose and Troubleshoot Computer Systems and Networks	Maintain Computer Systems and Networks	Operate a word-processing application	Operate a spreadsheet application	Operate a presentation package	Install and maintain a server
Design program logic	Apply program development approach	Apply object-oriented program language skills	Apply programming skills in a second language	Install network hardware to a network	Install software to networked computers	Install and configure a network
Determine and confirm client business expectations and needs	Create a simple mark-up language document to specification	Design a website to meet technical requirements	Transfer content to a website using commercial packages	Determine and apply appropriate development methodologies	Ensure website content meets technical protocols and standards	Build a database
Use structured query language to create database structures & manipulate data	Develop detailed technical design	Act on and complete change requests	Determine and act on client computing problems	Provide one -to - one instruction	Provide first-level remote help desk support	Design pages using a page layout application
Develop design studies	Create vector graphics using a graphics application	Create raster graphics using a graphics application	Develop designs for print media	Develop designs for electronic media	Develop designs for product packaging	Design booth & product/window display
Design program logic	Plan development of application features	Apply .Net programming skills	Develop Windows forms application	Develop web application	Develop service oriented applications	Develop applications using COBOL or similar language
Develop command-line/ console and desktop applications using Java technology	Develop enterprise/web applications using Java technology	Perform relational database management in Oracle database technology	Use and apply PL/SQL programming language	Design and Tune PL/SQL language	Perform object-oriented analysis and design in Java technology	Create and fine-tune Java technology applications using object-oriented programming concept

DEFINITION OF TERMS

GENERAL

- 1) **Certification** - is the process of verifying and validating the competencies of a person through assessment
- 2) **Certificate of Competency (COC)** – is a certification issued to individuals who pass the assessment for a single unit or cluster of units of competency
- 3) **Common Competencies** - are the skills and knowledge needed by all people working in a particular industry
- 4) **Competency** - is the possession and application of knowledge, skills and attitudes to perform work activities to the standard expected in the workplace
- 5) **Competency Assessment** - is the process of collecting evidence and making judgments on whether competency has been achieved
- 6) **Competency Standard (CS)** - is 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** - are the specific skills and knowledge needed in a particular area of work - industry sector/occupation/job role
- 9) **Critical aspects of competency** - refers to the evidence that is essential for successful performance of the unit of competency
- 10) **Elective Competencies** - are the additional skills and knowledge required by the individual or enterprise for work
- 11) **Elements** - are the building blocks of a unit of competency. They describe in outcome terms the functions that a person perform in the workplace
- 12) **Evidence Guide** - is a component of the unit of competency that defines or identifies the evidences 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
- 13) **Level** - refers to the category of skills and knowledge required to do a job
- 14) **Method of Assessment** - refers to the ways of collecting evidence and when evidence should be collected
- 15) **National Certificate (NC)** – is 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
- 16) **Performance Criteria** - are evaluative statements that specify what is to be assessed and the required level of performance
- 17) **Qualification** - is 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

- 18) **Range of Variables** - describes the circumstances or context in which the work is to be performed
- 19) **Recognition of Prior Learning (RPL)** – is the acknowledgement of an individual’s skills, knowledge and attitudes gained from life and work experiences outside registered training programs
- 20) **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
- 21) **Basic Competencies** - are the skills and knowledge that everyone needs for work
- 22) **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
- 23) **Underpinning Knowledge** - refers to the competency that involves in applying knowledge to perform work activities. It includes specific knowledge that is essential to the performance of the competency
- 24) **Underpinning Skills** - refers to the list of the skills needed to achieve the elements and performance criteria in the unit of competency. It includes generic and industry specific skills
- 25) **Unit of Competency** – is 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

1. **ADO.NET** - is a set of computer software components that programmers can use to access data and data services. It is a part of the base class library that is included with 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 non-relational 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.
2. **ASP.NET** - is 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.
3. **Algorithm** - is 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.
4. **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.

5. **Browser** – a software package that provides the user interface for accessing Internet, intranet and extranet Web sites.
6. **COBOL** - is one of the oldest programming languages. Its name is an acronym for **CO**mmun **B**usiness-**O**riented **L**anguage, defining its primary domain in business, finance, and administrative systems for companies and governments.
7. **Compiler** - is 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.
8. **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.
9. **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.
10. **Computer programming** - is 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.
11. **Computer Terminal** – any input/output device connected by telecommunications links to a computer.
12. **C language** - is 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.
13. **C++ language** - is 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.
14. **Data** - objective measurements of the attributes (characteristics) of entities such as people, places, things, and events.
15. **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.
16. **Decompiler** - is 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)
17. **Documentation** – a collection of documents or information.
18. **Edit** – to modify the form or format of data

19. **End user** – anyone who uses an information system or the information it produces.
20. **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.
21. **HTML**, which stands for HyperText Markup Language - is 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.
22. **Information** – data placed in a meaningful and useful context for an end user.
23. **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.
24. **Integrated development environment (IDE)** - is 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.
25. **Java** - is 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
26. **Java applications** - are typically compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture.
27. **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 work sites.
28. **Microsoft .NET Framework** - is 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.
29. **Microsoft SQL Server** - is a relational model database server produced by Microsoft. Its primary query languages are T-SQL and ANSI SQL
30. **Object code** - or an object file, is 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.

31. **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.
32. **Oracle Forms** - is 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.
33. **Oracle Reports** - is 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).
34. **Outsourcing** – turning over all or part of an organization’s information systems operation to outside contractors, known as systems integrators or facilities management companies.
35. **Programming language** - is 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.
36. **Quality Assurance** – methods for ensuring that information systems are free from errors and fraud and provide information products of high quality.
37. **Relational database management system (RDBMS)** - is 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.
38. **Service-oriented programming (SOP)** - is 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.”
39. **Software** – computer programs and procedures concerned with the operation of an information system.
40. **Source code** - is 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.
41. **SQL**, often referred to as **Structured Query Language** - is 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.
42. **Standards** – measures of performance developed to evaluate the progress of a system toward its objectives
43. **System** – an assembly of methods, procedures, or techniques unified by regulated interaction to form an organized whole
44. **User- friendly** – a characteristic of human-operated equipment and systems that makes them safe, comfortable, and easy to use.

45. **User interface** - is 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.
46. **VB.NET** – is a redesigned, object-oriented dialect of Visual Basic.
47. **Web application** - is 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.
48. **Windows Forms** - is 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.
49. **Windows Forms application** - is 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.

ACKNOWLEDGEMENTS

The Technical Education and Skills Development Authority (TESDA) wishes to extend thanks and appreciation to the many representatives of business, industry, academe and government agencies who donated their time and expertise to the development and validation of these Training Regulations.

- **THE TECHNICAL EXPERTS PANEL (TEP)**

Mr. William Lee	- Oracle Corp. Philippines
Mr. Nick Gazo	- GNA IT Consulting Services/Genix Labs
Ms. Geraldine Gazo	- GNA IT Consulting Services/Genix Labs
Mr. Calen Legaspi	- Orange and Bronze Software Labs

- **THE PARTICIPANTS IN THE NATIONAL VALIDATION OF THIS TRAINING REGULATION**

- Philippine Software Industries Association of the Philippines Members

- **THE TESDA BOARD - STANDARDS SETTING AND SYSTEMS DEVELOPMENT COMMITTEE**

- **THE MANAGEMENT AND STAFF OF THE TESDA SECRETARIAT**

- **Qualifications and Standards Office (QSO)**

- Competency Standards Division

Mr. Zoilo C. Galang	- CSD-QSO-TESDA
Mr. Samuel E. Calado Jr.	- CSD-QSO-TESDA
Mr. Stephen I. Cezar	- CSD-QSO-TESDA
Mr. Venzel Y. Concoles	- CSD-QSO-TESDA

- Curriculum and Training Aids Division

Mr. Arsenio A. Mateo, Jr.	- CTAD-QSO-TESDA
---------------------------	------------------