सत्यमेव जयते
GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT
& ENTREPRENEURSHIP

कौशल गुणवत्ता प्रगति

Qualification Pack

Jr. Embedded Developer

QP Code: MSME/ELE/Q1701

Version: 1.0

NSQF Level: 4.5

MSME TECHNOLOGY CENTRE ||
B-36 CHANDAKA INDUSTRIAL AREA || email:msmeexamcell@gmail.com

Contents

Qualification Pack

MSME/ELE/Q1701: Jr. Embedded Developer

Brief Job Description

Learners who attain this qualification will be able to Develop Programmes for Microcontrollers used in Industrial Applications

Personal Attributes

Learners who attain this qualification will be able to Develop Programmes for Microcontrollers used in Industrial Applications

Applicable National Occupational Standards (NOS)

Compulsory NOS:

1. MSME/ELE/N1702: Embedded Program Development on different microcontroller
2. MSME/ELE/N1701: Embedded Program Development on different microcontroller
3. MSME/ELE/N1706: Digital Logic design and Programming on FPGA and CPLD
4. MSME/ELE/N1705: Digital Logic design and Programming on FPGA and CPLD
5. MSME/ELE/N1704: IoT application development on R-Pi and Node-MCU
6. MSME/ELE/N1703: IoT application development on R-Pi and Node-MCU
7. MSME/ELE/N1709: PROJECT
8. MSME/ELE/N1708: Layout Design and Fabrication of PCB
9. MSME/ELE/N1707: Layout Design and Fabrication of PCB
10. MSME/ELE/N1710: Employability Skills

Qualification Pack (QP) Parameters

Sector	Electronics
Sub-Sector	Strategic Electronics
Occupation	Embedded Development

Qualification Pack

Country	India
NSQF Level	4.5
Credits	20
Aligned to NCO/ISCO/ISIC Code	(Embedded Software Engineer)
Minimum Educational Qualification & Experience	Completed 3-year diploma (after 10th) with NA of experience OR Pursuing 3rd year of 3-year diploma after 10th with NA of experience OR Certificate-NSQF (NSQF Level 4) with 1.5 years of experience
Minimum Level of Education for Training in School	10th Class
Pre-Requisite License or Training	NA
Minimum Job Entry Age	18 Years
Last Reviewed On	NA
Next Review Date	30/04/2027
NSQC Approval Date	30/04/2024
Version	1.0
Reference code on NQR	NCVET- QG-4.5-EH-02386-2024-V1-MSME
NQR Version	1.0

MSME/ELE/N1702: Embedded Program Development on different microcontroller

Description

After completion of course Student should be able to Understand the basic concept of C/C++ programming language

Scope

The scope covers the following :

- After completion of course Student should be able to Understand the basic concept of C/C++ programming language

Elements and Performance Criteria

MSME/AET/01 Embedded Program Development on different microcontroller

To be competent, the user/individual on the job must be able to:

PC1. • Explain the difference between a high-level programming • language and a low-level programming language

PC2. • What are the key features and characteristics of the C • programming language.

PC3. • What are decision-making statements in C, and why are they • important in programming.

PC4. • Differentiate between the if statement, the else-if statement, • and the switch statement in C

PC5. • Provide an example of a C program that uses a conditional • statement to determine if a number is even or odd

PC6. • Create a C program that uses a structure to represent a • student's information and a pointer to manipulate that • information.

PC7. • Explain the concept of pointers in C. How are they different • from regular variables

PC8. • Write a C program that calculates the average of an array of • numbers using a user-defined function.

PC9. • What is an array in C, and how does it differ from a regular • variable? Provide an example.

PC10. • What are peripherals in the context of microcontrollers and • embedded systems, and why are they essential.

PC11. • Write a simple Embedded C program that blinks an LED • connected to a PIC microcontroller. Explain the code logic.

PC12. • Configure and use a timer in a PIC microcontroller using • Embedded C code

PC13. • Write an Embedded C program that generates PWM signals to • control the brightness of an LED using a PIC microcontroller

PC14. • Provide an example of using a counter to count external events
• or pulses in a PIC microcontroller

PC15. • The key features and specifications of the LPC2148
• microcontroller, including its architecture and clock frequency.

PC16. Describe the characteristics of the ADC, DAC in LPC2148 including resolution and voltage output range

PC17. • Functioning of the UART/SPI/I2C communication protocols of
• the LPC2148 microcontroller.

PC18. Introduction to Cortex-M3 series

PC19. • Some common use cases or applications where the LPC1768
• microcontroller (Cortex-M3 series) is typically deployed

PC20. • The architecture of the Cortex-M3 core, focusing on its key
• components, such as the pipeline, registers, and execution
• modes.

PC21. • Examples of common peripherals that can be interfaced with
• the LPC1768, such as sensors, displays, and communication
• modules

PC22. • What is a real-time operating system (RTOS), and how does it
• differ from conventional operating systems

PC23. • Describe the concept of "real-time" in the context of RTOS.
• What are the different types of real-time systems

PC24. • How does an RTOS manage and schedule tasks or processes to
• meet specific timing constraints and deadlines

PC25. • What is the role of the kernel in an RTOS, and how does it differ
• from the kernel in a general-purpose OS

PC26. • Explain the significance of real-time operating systems in critical
• applications, such as aviation, automotive, and industrial
• control.

PC27. Compare and contrast the key differences between a generalpurpose operating system (OS) and an RTOS.

PC28. What are the strengths and limitations of using a generalpurpose OS in real-time applications versus an RTOS?

PC29. how task scheduling and resource management differ in an RTOS compared to a traditional OS

PC30. • Describe the various task scheduling methods employed in
• RTOS programming, including preemptive scheduling and
• cooperative scheduling.

PC31. • How does round-robin scheduling work in the context of an
• RTOS? What are the advantages and limitations of this
• approach

PC32. • Provide an example of a scenario where a priority-based
• scheduling policy is crucial in an RTOS application.

Assessment Criteria

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
MSME/AET/01 Embedded Program Development on different microcontroller	-	100	-	-
PC1. • Explain the difference between a high-level programming • language and a low-level programming language	-	-	-	-
PC2. • What are the key features and characteristics of the C • programming language.	-	-	-	-
PC3. • What are decision-making statements in C, and why are they • important in programming.	-	-	-	-
PC4. • Differentiate between the if statement, the else-if statement, • and the switch statement in C	-	-	-	-
PC5. • Provide an example of a C program that uses a conditional • statement to determine if a number is even or odd	-	-	-	-
PC6. • Create a C program that uses a structure to represent a • student's information and a pointer to manipulate that • information.	-	-	-	-
PC7. • Explain the concept of pointers in C. How are they different • from regular variables	-	-	-	-
PC8. • Write a C program that calculates the average of an array of • numbers using a user-defined function.	-	-	-	-

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
PC9. • What is an array in C, and how does it differ from a regular • variable? Provide an example.	-	-	-	-
PC10. • What are peripherals in the context of microcontrollers and • embedded systems, and why are they essential.	-	-	-	-
PC11. • Write a simple Embedded C program that blinks an LED • connected to a PIC microcontroller. Explain the code logic.	-	-	-	-
PC12. • Configure and use a timer in a PIC microcontroller using • Embedded C code	-	-	-	-
PC13. • Write an Embedded C program that generates PWM signals to • control the brightness of an LED using a PIC microcontroller	-	-	-	-
PC14. • Provide an example of using a counter to count external events • or pulses in a PIC microcontroller	-	-	-	-
PC15. • The key features and specifications of the LPC2148 • microcontroller, including its architecture and clock frequency.	-	-	-	-
PC16. Describe the characteristics of the ADC, DAC in LPC2148 including resolution and voltage output range	-	-	-	-
PC17. • Functioning of the UART/SPI/I2C communication protocols of • the LPC2148 microcontroller.	-	-	-	-
PC18. Introduction to Cortex-M3 series	-	-	-	-

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
PC19. • Some common use cases or applications where the LPC1768 • microcontroller (Cortex-M3 series) is typically deployed	-	-	-	-
PC20. • The architecture of the Cortex-M3 core, focusing on its key • components, such as the pipeline, registers, and execution • modes.	-	-	-	-
PC21. • Examples of common peripherals that can be interfaced with • the LPC1768, such as sensors, displays, and communication • modules	-	-	-	-
PC22. • What is a real-time operating system (RTOS), and how does it • differ from conventional operating systems	-	-	-	-
PC23. • Describe the concept of "real-time" in the context of RTOS. • What are the different types of real-time systems	-	-	-	-
PC24. • How does an RTOS manage and schedule tasks or processes to • meet specific timing constraints and deadlines	-	-	-	-
PC25. • What is the role of the kernel in an RTOS, and how does it differ • from the kernel in a general-purpose OS	-	-	-	-
PC26. • Explain the significance of real-time operating systems in critical • applications, such as aviation, automotive, and industrial • control.	-	-	-	-
PC27. Compare and contrast the key differences between a generalpurpose operating system (OS) and an RTOS.	-	-	-	-

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
PC28. What are the strengths and limitations of using a generalpurpose OS in real-time applications versus an RTOS?	-	-	-	-
PC29. how task scheduling and resource management differ in an RTOS compared to a traditional OS	-	-	-	-
PC30. • Describe the various task scheduling methods employed in • RTOS programming, including preemptive scheduling and • cooperative scheduling.	-	-	-	-
PC31. • How does round-robin scheduling work in the context of an • RTOS? What are the advantages and limitations of this • approach	-	-	-	-
PC32. • Provide an example of a scenario where a priority-based • scheduling policy is crucial in an RTOS application.	-	-	-	-
NOS Total	**-**	**100**	**-**	**-**

National Occupational Standards (NOS) Parameters

NOS Code	MSME/ELE/N1702
NOS Name	Embedded Program Development on different microcontroller
Sector	Electronics
Sub-Sector	
Occupation	Embedded Development
NSQF Level	4.5
Credits	4
Version	1.0
Last Reviewed Date	30/04/2024
Next Review Date	30/04/2027
NSQC Clearance Date	30/04/2024

MSME/ELE/N1701: Embedded Program Development on different microcontroller

Description

After completion of course Student should be able to Understand the basic concept of C/C++ programming language

Scope

The scope covers the following :

- After completion of course Student should be able to Understand the basic concept of C/C++ programming language

Elements and Performance Criteria

MSME/AET/01 Embedded Program Development on different microcontroller

To be competent, the user/individual on the job must be able to:

PC1. • Explain the difference between a high-level programming • language and a low-level programming language

PC2. • What are the key features and characteristics of the C • programming language.

PC3. • What are decision-making statements in C, and why are they • important in programming.

PC4. • Differentiate between the if statement, the else-if statement, • and the switch statement in C

PC5. • Provide an example of a C program that uses a conditional • statement to determine if a number is even or odd

PC6. • Create a C program that uses a structure to represent a • student's information and a pointer to manipulate that • information.

PC7. • Explain the concept of pointers in C. How are they different • from regular variables

PC8. • Write a C program that calculates the average of an array of • numbers using a user-defined function.

PC9. • What is an array in C, and how does it differ from a regular • variable? Provide an example.

PC10. • What are peripherals in the context of microcontrollers and • embedded systems, and why are they essential.

PC11. • Write a simple Embedded C program that blinks an LED • connected to a PIC microcontroller. Explain the code logic.

PC12. • Configure and use a timer in a PIC microcontroller using • Embedded C code

PC13. • Write an Embedded C program that generates PWM signals to • control the brightness of an LED using a PIC microcontroller

PC14. • Provide an example of using a counter to count external events
• or pulses in a PIC microcontroller

PC15. • The key features and specifications of the LPC2148
• microcontroller, including its architecture and clock frequency.

PC16. Describe the characteristics of the ADC, DAC in LPC2148 including resolution and voltage output range

PC17. • Functioning of the UART/SPI/I2C communication protocols of
• the LPC2148 microcontroller.

PC18. Introduction to Cortex-M3 series

PC19. • Some common use cases or applications where the LPC1768
• microcontroller (Cortex-M3 series) is typically deployed

PC20. • The architecture of the Cortex-M3 core, focusing on its key
• components, such as the pipeline, registers, and execution
• modes.

PC21. • Examples of common peripherals that can be interfaced with
• the LPC1768, such as sensors, displays, and communication
• modules

PC22. • What is a real-time operating system (RTOS), and how does it
• differ from conventional operating systems

PC23. • Describe the concept of "real-time" in the context of RTOS.
• What are the different types of real-time systems

PC24. • How does an RTOS manage and schedule tasks or processes to
• meet specific timing constraints and deadlines

PC25. • What is the role of the kernel in an RTOS, and how does it differ
• from the kernel in a general-purpose OS

PC26. • Explain the significance of real-time operating systems in critical
• applications, such as aviation, automotive, and industrial
• control.

PC27. Compare and contrast the key differences between a generalpurpose operating system (OS) and an RTOS.

PC28. What are the strengths and limitations of using a generalpurpose OS in real-time applications versus an RTOS?

PC29. how task scheduling and resource management differ in an RTOS compared to a traditional OS

PC30. • Describe the various task scheduling methods employed in
• RTOS programming, including preemptive scheduling and
• cooperative scheduling.

PC31. • How does round-robin scheduling work in the context of an
• RTOS? What are the advantages and limitations of this
• approach

PC32. • Provide an example of a scenario where a priority-based
• scheduling policy is crucial in an RTOS application.

Assessment Criteria

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
MSME/AET/01 Embedded Program Development on different microcontroller	**100**	**-**	**-**	**-**
PC1. • Explain the difference between a high-level programming • language and a low-level programming language	-	-	-	-
PC2. • What are the key features and characteristics of the C • programming language.	-	-	-	-
PC3. • What are decision-making statements in C, and why are they • important in programming.	-	-	-	-
PC4. • Differentiate between the if statement, the else-if statement, • and the switch statement in C	-	-	-	-
PC5. • Provide an example of a C program that uses a conditional • statement to determine if a number is even or odd	-	-	-	-
PC6. • Create a C program that uses a structure to represent a • student's information and a pointer to manipulate that • information.	-	-	-	-
PC7. • Explain the concept of pointers in C. How are they different • from regular variables	-	-	-	-
PC8. • Write a C program that calculates the average of an array of • numbers using a user-defined function.	-	-	-	-

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
PC9. • What is an array in C, and how does it differ from a regular • variable? Provide an example.	-	-	-	-
PC10. • What are peripherals in the context of microcontrollers and • embedded systems, and why are they essential.	-	-	-	-
PC11. • Write a simple Embedded C program that blinks an LED • connected to a PIC microcontroller. Explain the code logic.	-	-	-	-
PC12. • Configure and use a timer in a PIC microcontroller using • Embedded C code	-	-	-	-
PC13. • Write an Embedded C program that generates PWM signals to • control the brightness of an LED using a PIC microcontroller	-	-	-	-
PC14. • Provide an example of using a counter to count external events • or pulses in a PIC microcontroller	-	-	-	-
PC15. • The key features and specifications of the LPC2148 • microcontroller, including its architecture and clock frequency.	-	-	-	-
PC16. Describe the characteristics of the ADC, DAC in LPC2148 including resolution and voltage output range	-	-	-	-
PC17. • Functioning of the UART/SPI/I2C communication protocols of • the LPC2148 microcontroller.	-	-	-	-
PC18. Introduction to Cortex-M3 series	-	-	-	-

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
PC19. • Some common use cases or applications where the LPC1768 • microcontroller (Cortex-M3 series) is typically deployed	-	-	-	-
PC20. • The architecture of the Cortex-M3 core, focusing on its key • components, such as the pipeline, registers, and execution • modes.	-	-	-	-
PC21. • Examples of common peripherals that can be interfaced with • the LPC1768, such as sensors, displays, and communication • modules	-	-	-	-
PC22. • What is a real-time operating system (RTOS), and how does it • differ from conventional operating systems	-	-	-	-
PC23. • Describe the concept of "real-time" in the context of RTOS. • What are the different types of real-time systems	-	-	-	-
PC24. • How does an RTOS manage and schedule tasks or processes to • meet specific timing constraints and deadlines	-	-	-	-
PC25. • What is the role of the kernel in an RTOS, and how does it differ • from the kernel in a general-purpose OS	-	-	-	-
PC26. • Explain the significance of real-time operating systems in critical • applications, such as aviation, automotive, and industrial • control.	-	-	-	-
PC27. Compare and contrast the key differences between a generalpurpose operating system (OS) and an RTOS.	-	-	-	-

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
PC28. What are the strengths and limitations of using a generalpurpose OS in real-time applications versus an RTOS?	-	-	-	-
PC29. how task scheduling and resource management differ in an RTOS compared to a traditional OS	-	-	-	-
PC30. • Describe the various task scheduling methods employed in • RTOS programming, including preemptive scheduling and • cooperative scheduling.	-	-	-	-
PC31. • How does round-robin scheduling work in the context of an • RTOS? What are the advantages and limitations of this • approach	-	-	-	-
PC32. • Provide an example of a scenario where a priority-based • scheduling policy is crucial in an RTOS application.	-	-	-	-
NOS Total	**100**	**-**	**-**	**-**

National Occupational Standards (NOS) Parameters

NOS Code	MSME/ELE/N1701
NOS Name	Embedded Program Development on different microcontroller
Sector	Electronics
Sub-Sector	
Occupation	Embedded Development
NSQF Level	4.5
Credits	1
Version	1.0
Last Reviewed Date	30/04/2024
Next Review Date	30/04/2027
NSQC Clearance Date	30/04/2024

MSME/ELE/N1706: Digital Logic design and Programming on FPGA and CPLD

Description

After completion of course Student should be able to Ability to simulate using FPGA blocks.

Scope

The scope covers the following :

- After completion of course Student should be able to Ability to simulate using FPGA blocks.

Elements and Performance Criteria

MSME/AET/03 Digital Logic design and Programming on FPGA and CPLD

To be competent, the user/individual on the job must be able to:

PC1. • Explain the basic architecture of a CPLD (Complex
• Programmable Logic Device) and an FPGA (Field-Programmable
• Gate Array). What differentiates these two devices

PC2. • Using Xilinx ISE (Integrated Software Environment), describe the
• steps involved in creating a simple project for a CPLD or FPGA

PC3. • Develop a VHDL program for a full adder circuit and explain the
• key components of the code.

PC4. • Explain the concept of memory blocks in system architecture.
• How do these blocks store and retrieve data

PC5. • Provide an overview of the different types of interfaces found in
• embedded systems, such as UART, SPI, I2C, and GPIO.

PC6. • How do domain-specific languages (DSLs) differ from generalpurpose languages? Provide examples of DSLs and their use
• cases

PC7. • VHDL code snippet for a simple digital circuit, such as a 2-to-1
• multiplexer.

PC8. • How do you read and interpret HDL or VHDL code for a specific
• digital design? Describe the syntax and structure of such code

PC9. • How do these tools facilitate various stages of the design
• process, from conceptualization to final verification?

Assessment Criteria

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
MSME/AET/03 Digital Logic design and Programming on FPGA and CPLD	-	**100**	-	-
PC1. • Explain the basic architecture of a CPLD (Complex • Programmable Logic Device) and an FPGA (Field-Programmable • Gate Array). What differentiates these two devices	-	-	-	-
PC2. • Using Xilinx ISE (Integrated Software Environment), describe the • steps involved in creating a simple project for a CPLD or FPGA	-	-	-	-
PC3. • Develop a VHDL program for a full adder circuit and explain the • key components of the code.	-	-	-	-
PC4. • Explain the concept of memory blocks in system architecture. • How do these blocks store and retrieve data	-	-	-	-
PC5. • Provide an overview of the different types of interfaces found in • embedded systems, such as UART, SPI, I2C, and GPIO.	-	-	-	-
PC6. • How do domain-specific languages (DSLs) differ from generalpurpose languages? Provide examples of DSLs and their use • cases	-	-	-	-
PC7. • VHDL code snippet for a simple digital circuit, such as a 2-to-1 • multiplexer.	-	-	-	-

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
PC8. • How do you read and interpret HDL or VHDL code for a specific • digital design? Describe the syntax and structure of such code	-	-	-	-
PC9. • How do these tools facilitate various stages of the design • process, from conceptualization to final verification?	-	-	-	-
NOS Total	**-**	**100**	**-**	**-**

National Occupational Standards (NOS) Parameters

NOS Code	MSME/ELE/N1706
NOS Name	Digital Logic design and Programming on FPGA and CPLD
Sector	Electronics
Sub-Sector	
Occupation	Embedded Development
NSQF Level	4.5
Credits	2
Version	1.0
Last Reviewed Date	30/04/2024
Next Review Date	30/04/2027
NSQC Clearance Date	30/04/2024

MSME/ELE/N1705: Digital Logic design and Programming on FPGA and CPLD

Description

After completion of course Student should be able to Ability to simulate using FPGA blocks.

Scope

The scope covers the following :

- After completion of course Student should be able to Ability to simulate using FPGA blocks.

Elements and Performance Criteria

MSME/AET/03 Digital Logic design and Programming on FPGA and CPLD

To be competent, the user/individual on the job must be able to:

PC1. • Explain the basic architecture of a CPLD (Complex
• Programmable Logic Device) and an FPGA (Field-Programmable
• Gate Array). What differentiates these two devices

PC2. • Using Xilinx ISE (Integrated Software Environment), describe the
• steps involved in creating a simple project for a CPLD or FPGA

PC3. • Develop a VHDL program for a full adder circuit and explain the
• key components of the code.

PC4. • Explain the concept of memory blocks in system architecture.
• How do these blocks store and retrieve data

PC5. • Provide an overview of the different types of interfaces found in
• embedded systems, such as UART, SPI, I2C, and GPIO.

PC6. • How do domain-specific languages (DSLs) differ from generalpurpose languages? Provide examples of DSLs and their use
• cases

PC7. • VHDL code snippet for a simple digital circuit, such as a 2-to-1
• multiplexer.

PC8. • How do you read and interpret HDL or VHDL code for a specific
• digital design? Describe the syntax and structure of such code

PC9. • How do these tools facilitate various stages of the design
• process, from conceptualization to final verification?

Assessment Criteria

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
MSME/AET/03 Digital Logic design and Programming on FPGA and CPLD	**100**	**-**	**-**	**-**
PC1. • Explain the basic architecture of a CPLD (Complex • Programmable Logic Device) and an FPGA (Field-Programmable • Gate Array). What differentiates these two devices	-	-	-	-
PC2. • Using Xilinx ISE (Integrated Software Environment), describe the • steps involved in creating a simple project for a CPLD or FPGA	-	-	-	-
PC3. • Develop a VHDL program for a full adder circuit and explain the • key components of the code.	-	-	-	-
PC4. • Explain the concept of memory blocks in system architecture. • How do these blocks store and retrieve data	-	-	-	-
PC5. • Provide an overview of the different types of interfaces found in • embedded systems, such as UART, SPI, I2C, and GPIO.	-	-	-	-
PC6. • How do domain-specific languages (DSLs) differ from generalpurpose languages? Provide examples of DSLs and their use • cases	-	-	-	-
PC7. • VHDL code snippet for a simple digital circuit, such as a 2-to-1 • multiplexer.	-	-	-	-

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
PC8. • How do you read and interpret HDL or VHDL code for a specific • digital design? Describe the syntax and structure of such code	-	-	-	-
PC9. • How do these tools facilitate various stages of the design • process, from conceptualization to final verification?	-	-	-	-
NOS Total	**100**	**-**	**-**	**-**

National Occupational Standards (NOS) Parameters

NOS Code	MSME/ELE/N1705
NOS Name	Digital Logic design and Programming on FPGA and CPLD
Sector	Electronics
Sub-Sector	
Occupation	Embedded Development
NSQF Level	4.5
Credits	1
Version	1.0
Last Reviewed Date	30/04/2024
Next Review Date	30/04/2027
NSQC Clearance Date	30/04/2024

Qualification Pack

MSME/ELE/N1704: IoT application development on R-Pi and Node-MCU

Description

After completion of course Student should be able to Introduction to Python Programming

Scope

The scope covers the following :

- After completion of course Student should be able to Introduction to Python Programming

Elements and Performance Criteria

MSME/AET/02 IoT application development on R-Pi and NodeMCU

To be competent, the user/individual on the job must be able to:

PC1. • What is the Internet of Things (IoT), and how does it extend the • capabilities of connected devices

PC2. • The basic concepts of programming and how Python fits into • the realm of computer programming

PC3. • The different methods of decision-making in Python, including if • statements and switch/case constructs

PC4. • Loops like for and while work in Python, and what are their use • cases

PC5. • The Python environment, including the interpreter and how it • executes Python code

PC6. • The concept of looping structures in Python and how they • enable repetitive tasks

PC7. • The use and characteristics of data structures like lists, tuples, • sets, and dictionaries in Python.

PC8. • The concept of file handling in Python, including, appending, • reading and writing to files

PC9. Provide code examples demonstrating the use of constructors and destructors.

PC10. • Concept of GPIO (General Purpose Input/output) pins and how • they can be used to interface with external devices for program • execution

PC11. • Concept of GPIO (General Purpose Input/output) pins and how • they can be used to interface with external devices for program • execution

PC12. • Example of using port programming to toggle an LED connected • to a GPIO pin

PC13. • Process of programming the Raspberry Pi to control an LED, • LCD, sensors, Actuators etc

PC14. • Adding a Bluetooth functionality to a Raspberry Pi, and what • types of Bluetooth modules are commonly used

PC15. • Explain the core components of an IoT system, including • sensors, communication protocols, and data processing

PC16. • What factors should be considered when choosing an IoT
• platform for a specific project

PC17. • Example of using the ThingSpeak API to collect and analyze
• sensor data from an IoT device.

PC18. The concept of interfacing IoT devices, like Arduino, with cloudbased web services and databases

PC19. • What is NodeMCU, and how does it differ from other IoT
• development platforms like Arduino

PC20. • How can various sensors, such as temperature, humidity,
• motion, ultrasonic, infrared, current, sound, and gas sensors, be
• interfaced with an Arduino board for IoT applications

PC21. • How are actuators like relay switches, motors interfaced and
• controlled using an Arduino board

Assessment Criteria

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
MSME/AET/02 IoT application development on R-Pi and NodeMCU	-	**100**	-	-
PC1. • What is the Internet of Things (IoT), and how does it extend the • capabilities of connected devices	-	-	-	-
PC2. • The basic concepts of programming and how Python fits into • the realm of computer programming	-	-	-	-
PC3. • The different methods of decision-making in Python, including if • statements and switch/case constructs	-	-	-	-
PC4. • Loops like for and while work in Python, and what are their use • cases	-	-	-	-
PC5. • The Python environment, including the interpreter and how it • executes Python code	-	-	-	-
PC6. • The concept of looping structures in Python and how they • enable repetitive tasks	-	-	-	-
PC7. • The use and characteristics of data structures like lists, tuples, • sets, and dictionaries in Python.	-	-	-	-
PC8. • The concept of file handling in Python, including, appending, • reading and writing to files	-	-	-	-
PC9. Provide code examples demonstrating the use of constructors and destructors.	-	-	-	-

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
PC10. • Concept of GPIO (General Purpose Input/output) pins and how • they can be used to interface with external devices for program • execution	-	-	-	-
PC11. • Concept of GPIO (General Purpose Input/output) pins and how • they can be used to interface with external devices for program • execution	-	-	-	-
PC12. • Example of using port programming to toggle an LED connected • to a GPIO pin	-	-	-	-
PC13. • Process of programming the Raspberry Pi to control an LED, • LCD, sensors, Actuators etc	-	-	-	-
PC14. • Adding a Bluetooth functionality to a Raspberry Pi, and what • types of Bluetooth modules are commonly used	-	-	-	-
PC15. • Explain the core components of an IoT system, including • sensors, communication protocols, and data processing	-	-	-	-
PC16. • What factors should be considered when choosing an IoT • platform for a specific project	-	-	-	-
PC17. • Example of using the ThingSpeak API to collect and analyze • sensor data from an IoT device.	-	-	-	-
PC18. The concept of interfacing IoT devices, like Arduino, with cloudbased web services and databases	-	-	-	-

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
PC19. • What is NodeMCU, and how does it differ from other IoT • development platforms like Arduino	-	-	-	-
PC20. • How can various sensors, such as temperature, humidity, • motion, ultrasonic, infrared, current, sound, and gas sensors, be • interfaced with an Arduino board for IoT applications	-	-	-	-
PC21. • How are actuators like relay switches, motors interfaced and • controlled using an Arduino board	-	-	-	-
NOS Total	**-**	**100**	**-**	**-**

National Occupational Standards (NOS) Parameters

NOS Code	MSME/ELE/N1704
NOS Name	IoT application development on R-Pi and Node-MCU
Sector	Electronics
Sub-Sector	
Occupation	Embedded Development
NSQF Level	4.5
Credits	2
Version	1.0
Last Reviewed Date	30/04/2024
Next Review Date	30/04/2027
NSQC Clearance Date	30/04/2024

Qualification Pack

MSME/ELE/N1703: IoT application development on R-Pi and Node-MCU

Description

After completion of course Student should be able to Introduction to Python Programming

Scope

The scope covers the following :

- After completion of course Student should be able to Introduction to Python Programming

Elements and Performance Criteria

MSME/AET/02 IoT application development on R-Pi and NodeMCU

To be competent, the user/individual on the job must be able to:

PC1. • What is the Internet of Things (IoT), and how does it extend the • capabilities of connected devices

PC2. • The basic concepts of programming and how Python fits into • the realm of computer programming

PC3. • The different methods of decision-making in Python, including if • statements and switch/case constructs

PC4. • Loops like for and while work in Python, and what are their use • cases

PC5. • The Python environment, including the interpreter and how it • executes Python code

PC6. • The concept of looping structures in Python and how they • enable repetitive tasks

PC7. • The use and characteristics of data structures like lists, tuples, • sets, and dictionaries in Python.

PC8. • The concept of file handling in Python, including, appending, • reading and writing to files

PC9. Provide code examples demonstrating the use of constructors and destructors.

PC10. • Concept of GPIO (General Purpose Input/output) pins and how • they can be used to interface with external devices for program • execution

PC11. • Concept of GPIO (General Purpose Input/output) pins and how • they can be used to interface with external devices for program • execution

PC12. • Example of using port programming to toggle an LED connected • to a GPIO pin

PC13. • Process of programming the Raspberry Pi to control an LED, • LCD, sensors, Actuators etc

PC14. • Adding a Bluetooth functionality to a Raspberry Pi, and what • types of Bluetooth modules are commonly used

PC15. • Explain the core components of an IoT system, including • sensors, communication protocols, and data processing

PC16. • What factors should be considered when choosing an IoT
• platform for a specific project

PC17. • Example of using the ThingSpeak API to collect and analyze
• sensor data from an IoT device.

PC18. The concept of interfacing IoT devices, like Arduino, with cloudbased web services and databases

PC19. • What is NodeMCU, and how does it differ from other IoT
• development platforms like Arduino

PC20. • How can various sensors, such as temperature, humidity,
• motion, ultrasonic, infrared, current, sound, and gas sensors, be
• interfaced with an Arduino board for IoT applications

PC21. • How are actuators like relay switches, motors interfaced and
• controlled using an Arduino board

Assessment Criteria

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
MSME/AET/02 IoT application development on R-Pi and NodeMCU	**100**	**-**	**-**	**-**
PC1. • What is the Internet of Things (IoT), and how does it extend the • capabilities of connected devices	-	-	-	-
PC2. • The basic concepts of programming and how Python fits into • the realm of computer programming	-	-	-	-
PC3. • The different methods of decision-making in Python, including if • statements and switch/case constructs	-	-	-	-
PC4. • Loops like for and while work in Python, and what are their use • cases	-	-	-	-
PC5. • The Python environment, including the interpreter and how it • executes Python code	-	-	-	-
PC6. • The concept of looping structures in Python and how they • enable repetitive tasks	-	-	-	-
PC7. • The use and characteristics of data structures like lists, tuples, • sets, and dictionaries in Python.	-	-	-	-
PC8. • The concept of file handling in Python, including, appending, • reading and writing to files	-	-	-	-
PC9. Provide code examples demonstrating the use of constructors and destructors.	-	-	-	-

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
PC10. • Concept of GPIO (General Purpose Input/output) pins and how • they can be used to interface with external devices for program • execution	-	-	-	-
PC11. • Concept of GPIO (General Purpose Input/output) pins and how • they can be used to interface with external devices for program • execution	-	-	-	-
PC12. • Example of using port programming to toggle an LED connected • to a GPIO pin	-	-	-	-
PC13. • Process of programming the Raspberry Pi to control an LED, • LCD, sensors, Actuators etc	-	-	-	-
PC14. • Adding a Bluetooth functionality to a Raspberry Pi, and what • types of Bluetooth modules are commonly used	-	-	-	-
PC15. • Explain the core components of an IoT system, including • sensors, communication protocols, and data processing	-	-	-	-
PC16. • What factors should be considered when choosing an IoT • platform for a specific project	-	-	-	-
PC17. • Example of using the ThingSpeak API to collect and analyze • sensor data from an IoT device.	-	-	-	-
PC18. The concept of interfacing IoT devices, like Arduino, with cloudbased web services and databases	-	-	-	-

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
PC19. • What is NodeMCU, and how does it differ from other IoT • development platforms like Arduino	-	-	-	-
PC20. • How can various sensors, such as temperature, humidity, • motion, ultrasonic, infrared, current, sound, and gas sensors, be • interfaced with an Arduino board for IoT applications	-	-	-	-
PC21. • How are actuators like relay switches, motors interfaced and • controlled using an Arduino board	-	-	-	-
NOS Total	**100**	**-**	**-**	**-**

National Occupational Standards (NOS) Parameters

NOS Code	MSME/ELE/N1703
NOS Name	IoT application development on R-Pi and Node-MCU
Sector	Electronics
Sub-Sector	
Occupation	Embedded Development
NSQF Level	4.5
Credits	1
Version	1.0
Last Reviewed Date	30/04/2024
Next Review Date	30/04/2027
NSQC Clearance Date	30/04/2024

MSME/ELE/N1709: PROJECT

Description

After completion of course Student should be able to Development of desired circuit schematics and printed circuit boards

Scope

The scope covers the following :

- After completion of course Student should be able to Development of desired circuit schematics and printed circuit boards

Elements and Performance Criteria

MSME/AET/ 06 PROJECT

To be competent, the user/individual on the job must be able to:

PC1. -NIL-

Assessment Criteria

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
MSME/AET/ 06 PROJECT	-	100	-	-
PC1. -NIL-	-	-	-	-
NOS Total	-	100	-	-

National Occupational Standards (NOS) Parameters

NOS Code	MSME/ELE/N1709
NOS Name	PROJECT
Sector	Electronics
Sub-Sector	
Occupation	Embedded Development
NSQF Level	4.5
Credits	4
Version	1.0
Last Reviewed Date	30/04/2024
Next Review Date	30/04/2027
NSQC Clearance Date	30/04/2024

Qualification Pack

MSME/ELE/N1708: Layout Design and Fabrication of PCB

Description

After completion of course Student should be able to Developing schematic and creating PCB layout, converting them to Gerber format using

Scope

The scope covers the following :

- After completion of course Student should be able to Developing schematic and creating PCB layout, converting them to Gerber format using

Elements and Performance Criteria

MSME/AET/04 Layout Design and Fabrication of PCB

To be competent, the user/individual on the job must be able to:

PC1. • What is the purpose of creating schematics in PCB design? How • do you choose the components and connections in a schematic

PC2. • How do you translate a schematic diagram into a physical PCB • layout? What software tools are commonly used for this process

PC3. • Discuss the considerations and challenges in arranging • components and routing traces on the PCB layout

PC4. • How does component placement impact the overall • performance and reliability of a PCB

PC5. • Compare and contrast the design considerations for single-layer • and multilayer PCBs. What are the advantages of each type

PC6. • What types of files are typically generated for PCB • manufacturing, and what is the purpose of each file format

PC7. • What are the key considerations when planning the design of a • circuit schematic before putting it on paper or into software

PC8. • Explain the steps involved in creating a new project schematic • using PCB design software. What software options are • commonly used for this purpose

PC9. • Explain the methods and tools used to perform a • comprehensive check on a schematic design.

PC10. • How do you ensure that the trace widths, pad sizes, and via • dimensions in the schematic align with the original PCB design • requirements

Assessment Criteria

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
MSME/AET/04 Layout Design and Fabrication of PCB	-	100	-	-
PC1. • What is the purpose of creating schematics in PCB design? How • do you choose the components and connections in a schematic	-	-	-	-
PC2. • How do you translate a schematic diagram into a physical PCB • layout? What software tools are commonly used for this process	-	-	-	-
PC3. • Discuss the considerations and challenges in arranging • components and routing traces on the PCB layout	-	-	-	-
PC4. • How does component placement impact the overall • performance and reliability of a PCB	-	-	-	-
PC5. • Compare and contrast the design considerations for single-layer • and multilayer PCBs. What are the advantages of each type	-	-	-	-
PC6. • What types of files are typically generated for PCB • manufacturing, and what is the purpose of each file format	-	-	-	-
PC7. • What are the key considerations when planning the design of a • circuit schematic before putting it on paper or into software	-	-	-	-

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
PC8. • Explain the steps involved in creating a new project schematic • using PCB design software. What software options are • commonly used for this purpose	-	-	-	-
PC9. • Explain the methods and tools used to perform a • comprehensive check on a schematic design.	-	-	-	-
PC10. • How do you ensure that the trace widths, pad sizes, and via • dimensions in the schematic align with the original PCB design • requirements	-	-	-	-
NOS Total	**-**	**100**	**-**	**-**

National Occupational Standards (NOS) Parameters

NOS Code	MSME/ELE/N1708
NOS Name	Layout Design and Fabrication of PCB
Sector	Electronics
Sub-Sector	
Occupation	Embedded Development
NSQF Level	4.5
Credits	2
Version	1.0
Last Reviewed Date	30/04/2024
Next Review Date	30/04/2027
NSQC Clearance Date	30/04/2024

MSME/ELE/N1707: Layout Design and Fabrication of PCB

Description

After completion of course Student should be able to Developing schematic and creating PCB layout, converting them to Gerber format using

Scope

The scope covers the following :

- After completion of course Student should be able to Developing schematic and creating PCB layout, converting them to Gerber format using

Elements and Performance Criteria

MSME/AET/04 Layout Design and Fabrication of PCB

To be competent, the user/individual on the job must be able to:

PC1. • What is the purpose of creating schematics in PCB design? How • do you choose the components and connections in a schematic

PC2. • How do you translate a schematic diagram into a physical PCB • layout? What software tools are commonly used for this process

PC3. • Discuss the considerations and challenges in arranging • components and routing traces on the PCB layout

PC4. • How does component placement impact the overall • performance and reliability of a PCB

PC5. • Compare and contrast the design considerations for single-layer • and multilayer PCBs. What are the advantages of each type

PC6. • What types of files are typically generated for PCB • manufacturing, and what is the purpose of each file format

PC7. • What are the key considerations when planning the design of a • circuit schematic before putting it on paper or into software

PC8. • Explain the steps involved in creating a new project schematic • using PCB design software. What software options are • commonly used for this purpose

PC9. • Explain the methods and tools used to perform a • comprehensive check on a schematic design.

PC10. • How do you ensure that the trace widths, pad sizes, and via • dimensions in the schematic align with the original PCB design • requirements

Assessment Criteria

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
MSME/AET/04 Layout Design and Fabrication of PCB	**100**	**-**	**-**	**-**
PC1. • What is the purpose of creating schematics in PCB design? How • do you choose the components and connections in a schematic	-	-	-	-
PC2. • How do you translate a schematic diagram into a physical PCB • layout? What software tools are commonly used for this process	-	-	-	-
PC3. • Discuss the considerations and challenges in arranging • components and routing traces on the PCB layout	-	-	-	-
PC4. • How does component placement impact the overall • performance and reliability of a PCB	-	-	-	-
PC5. • Compare and contrast the design considerations for single-layer • and multilayer PCBs. What are the advantages of each type	-	-	-	-
PC6. • What types of files are typically generated for PCB • manufacturing, and what is the purpose of each file format	-	-	-	-
PC7. • What are the key considerations when planning the design of a • circuit schematic before putting it on paper or into software	-	-	-	-

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
PC8. • Explain the steps involved in creating a new project schematic • using PCB design software. What software options are • commonly used for this purpose	-	-	-	-
PC9. • Explain the methods and tools used to perform a • comprehensive check on a schematic design.	-	-	-	-
PC10. • How do you ensure that the trace widths, pad sizes, and via • dimensions in the schematic align with the original PCB design • requirements	-	-	-	-
NOS Total	**100**	**-**	**-**	**-**

National Occupational Standards (NOS) Parameters

NOS Code	MSME/ELE/N1707
NOS Name	Layout Design and Fabrication of PCB
Sector	Electronics
Sub-Sector	
Occupation	Embedded Development
NSQF Level	4.5
Credits	1
Version	1.0
Last Reviewed Date	30/04/2024
Next Review Date	30/04/2027
NSQC Clearance Date	30/04/2024

MSME/ELE/N1710: Employability Skills

Description

This NOS unit is about carrying out operations about learners applying basic and advanced Employability Skills concepts in real life situations to become a successful 21st century professional

Scope

The scope covers the following :

- The scope covers the following:
- plan and prepare advance employability skills activities
- carry out the work to plan and prepare the learners to build key knowledge and skills for career
- dEvelopment in the 21st century using advanced employability skills
- documenting the record

Elements and Performance Criteria

MSME/ES/01 Employability Skills

To be competent, the user/individual on the job must be able to:

PC1. • Understand the significance of employability skills in meeting the job • requirements

PC2. • Identify constitutional values, civic rights, duties, personal values and • ethics and environmentally sustainable practices.

PC3. • Explain 21st Century Skills such as Self-Awareness, Behavior Skills, • Positive attitude, self-motivation, problem-solving, creative thinking, • time management, social and cultural awareness, emotional • awareness, continuous learning mindset etc

PC4. Speak with others using some basic English phrases or sentences

PC5. Follow good manners while communicating with others

PC6. Work with others in a team

PC7. Communicate and behave appropriately with all genders and PwD

PC8. Report any issues related to sexual harassment

PC9. Use various financial products and services safely and securely

PC10. Calculate income, expenses, savings etc.

PC11. • Approach the concerned authorities for any exploitation as per legal • rights and laws

PC12. • Operate digital devices and use its features and applications securely • and safely

PC13. Use internet and social media platforms securely and safely

PC14. Identify and assess opportunities for potential business

PC15. • Identify sources for arranging money and associated financial and legal • challenges

PC16. Identify different types of customers

PC17. Identify customer needs and address them appropriately
PC18. Follow appropriate hygiene and grooming standards
PC19. Create a basic biodata
PC20. Search for suitable jobs and apply
PC21. Identify and register apprenticeship opportunities as per requirement

Assessment Criteria

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
MSME/ES/01 Employability Skills	**100**	**-**	**-**	**-**
PC1. • Understand the significance of employability skills in meeting the job • requirements	-	-	-	-
PC2. • Identify constitutional values, civic rights, duties, personal values and • ethics and environmentally sustainable practices.	-	-	-	-
PC3. • Explain 21st Century Skills such as Self-Awareness, Behavior Skills, • Positive attitude, self-motivation, problem-solving, creative thinking, • time management, social and cultural awareness, emotional • awareness, continuous learning mindset etc	-	-	-	-
PC4. Speak with others using some basic English phrases or sentences	-	-	-	-
PC5. Follow good manners while communicating with others	-	-	-	-
PC6. Work with others in a team	-	-	-	-
PC7. Communicate and behave appropriately with all genders and PwD	-	-	-	-
PC8. Report any issues related to sexual harassment	-	-	-	-
PC9. Use various financial products and services safely and securely	-	-	-	-
PC10. Calculate income, expenses, savings etc.	-	-	-	-
PC11. • Approach the concerned authorities for any exploitation as per legal • rights and laws	-	-	-	-

Assessment Criteria for Outcomes	Theory Marks	Practical Marks	Project Marks	Viva Marks
PC12. • Operate digital devices and use its features and applications securely • and safely	-	-	-	-
PC13. Use internet and social media platforms securely and safely	-	-	-	-
PC14. Identify and assess opportunities for potential business	-	-	-	-
PC15. • Identify sources for arranging money and associated financial and legal • challenges	-	-	-	-
PC16. Identify different types of customers	-	-	-	-
PC17. Identify customer needs and address them appropriately	-	-	-	-
PC18. Follow appropriate hygiene and grooming standards	-	-	-	-
PC19. Create a basic biodata	-	-	-	-
PC20. Search for suitable jobs and apply	-	-	-	-
PC21. Identify and register apprenticeship opportunities as per requirement	-	-	-	-
NOS Total	**100**	**-**	**-**	**-**

National Occupational Standards (NOS) Parameters

NOS Code	MSME/ELE/N1710
NOS Name	Employability Skills
Sector	Electronics
Sub-Sector	
Occupation	Embedded Development
NSQF Level	4.5
Credits	2
Version	1.0
Last Reviewed Date	30/04/2024
Next Review Date	30/04/2027
NSQC Clearance Date	30/04/2024

Assessment Guidelines and Assessment Weightage

Assessment Guidelines

As per QP

Minimum Aggregate Passing % at QP Level : 40

(**Please note**: Every Trainee should score a minimum aggregate passing percentage as specified above, to successfully clear the Qualification Pack assessment.)

Minimum Passing % at NOS Level: 40

(**Please note**: A Trainee must score the minimum percentage for each NOS separately as well as on the QP as a whole.)

Assessment Weightage

Compulsory NOS

Qualification Pack

National Occupational Standards	Theory Marks	Practical Marks	Project Marks	Viva Marks	Total Marks	Weightage
MSME/ELE/N1702.Embedded Program Development on different microcontroller	-	100	-	-	100	10
MSME/ELE/N1701.Embedded Program Development on different microcontroller	100	-	-	-	100	10
MSME/ELE/N1706.Digital Logic design and Programming on FPGA and CPLD	-	100	-	-	100	10
MSME/ELE/N1705.Digital Logic design and Programming on FPGA and CPLD	100	-	-	-	100	10
MSME/ELE/N1704.IoT application development on R-Pi and Node-MCU	-	100	-	-	100	10
MSME/ELE/N1703.IoT application development on R-Pi and Node-MCU	100	-	-	-	100	10
MSME/ELE/N1709.PROJECT	-	100	-	-	100	10
MSME/ELE/N1708.Layout Design and Fabrication of PCB	-	100	-	-	100	10
MSME/ELE/N1707.Layout Design and Fabrication of PCB	100	-	-	-	100	10
MSME/ELE/N1710.Employability Skills	100	-	-	-	100	10
Total	**500**	**500**	**-**	**-**	**1000**	**100**

Acronyms

NOS	National Occupational Standard(s)
NSQF	National Skills Qualifications Framework
QP	Qualifications Pack
TVET	Technical and Vocational Education and Training

Glossary

Sector	Sector is a conglomeration of different business operations having similar business and interests. It may also be defined as a distinct subset of the economy whose components share similar characteristics and interests.
Sub-sector	Sub-sector is derived from a further breakdown based on the characteristics and interests of its components.
Occupation	Occupation is a set of job roles, which perform similar/ related set of functions in an industry.
Job role	Job role defines a unique set of functions that together form a unique employment opportunity in an organisation.
Occupational Standards (OS)	OS specify the standards of performance an individual must achieve when carrying out a function in the workplace, together with the Knowledge and Understanding (KU) they need to meet that standard consistently. Occupational Standards are applicable both in the Indian and global contexts.
Performance Criteria (PC)	Performance Criteria (PC) are statements that together specify the standard of performance required when carrying out a task.
National Occupational Standards (NOS)	NOS are occupational standards which apply uniquely in the Indian context.
Qualifications Pack (QP)	QP comprises the set of OS, together with the educational, training and other criteria required to perform a job role. A QP is assigned a unique qualifications pack code.
Unit Code	Unit code is a unique identifier for an Occupational Standard, which is denoted by an 'N'
Unit Title	Unit title gives a clear overall statement about what the incumbent should be able to do.
Description	Description gives a short summary of the unit content. This would be helpful to anyone searching on a database to verify that this is the appropriate OS they are looking for.
Scope	Scope is a set of statements specifying the range of variables that an individual may have to deal with in carrying out the function which have a critical impact on quality of performance required.

Knowledge and Understanding (KU)	Knowledge and Understanding (KU) are statements which together specify the technical, generic, professional and organisational specific knowledge that an individual needs in order to perform to the required standard.
Organisational Context	Organisational context includes the way the organisation is structured and how it operates, including the extent of operative knowledge managers have of their relevant areas of responsibility.
Technical Knowledge	Technical knowledge is the specific knowledge needed to accomplish specific designated responsibilities.
Core Skills/ Generic Skills (GS)	Core skills or Generic Skills (GS) are a group of skills that are the key to learning and working in today's world. These skills are typically needed in any work environment in today's world. These skills are typically needed in any work environment. In the context of the OS, these include communication related skills that are applicable to most job roles.
Electives	Electives are NOS/set of NOS that are identified by the sector as contributive to specialization in a job role. There may be multiple electives within a QP for each specialized job role. Trainees must select at least one elective for the successful completion of a QP with Electives.
Options	Options are NOS/set of NOS that are identified by the sector as additional skills. There may be multiple options within a QP. It is not mandatory to select any of the options to complete a QP with Options.