**900220-000-00-PM-02, Programming with C++, NQF Level 4, Credits 8**

**PRACTICAL MODULE (2)**

|  |  |
| --- | --- |
| **Module #** |  |
| **NQF Level** |  |
| **Occupational Code** |  |
| **SAQA QUAL ID** |  |
| **Qualification Title** |  |

**INTRODUCTION**

Congratulations on completing the program. As part of your training, you are required to keep a logbook of all practical on the job training and exposure you receive during the learning process.

You will now be assigned to a mentor who will oversee your off-site training, usually referred to as on-the-job training.

The mentor will assist and advise you on the practical aspects of the job, how to fit into the company, what is expected of you as an employee and as a future supervisor.

**Responsibilities of the learner include:**

* One hundred percent commitment to the learning process. Learners are encouraged to study any additional source of information relevant to this learning process.
* Doing all assignments contained in this workbook as well any tasks and assignments received from your mentor or supervisor to whom you have been assigned.
* Although the mentor is responsible to sign off all sections completed, it is the learner’s responsibility to ensure that all paperwork is completed and handed in for filing on his/her record of learning. It should be clearly stated to learners that a 100% complete record of learning, as prescribed by this logbook, is their sole responsibility. Any document missing from the record may result in your not being declared competent.
* Discuss any problems that you may have with your mentor.

***Acknowledgment of Receipt***

I \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Learner) acknowledge receipt of my on the job Training

Logbook on this the \_\_\_\_\_\_\_\_\_\_ day of \_\_\_\_\_\_\_\_\_\_\_\_ 200 \_\_\_

The process of on-the-job training has been explained to me.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of Learner

Name of Mentor: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of Mentor

**Declaration of Authenticity**

I (*learner name and surname*) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, ID Nr

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ hereby declare that the work contained herein was completed by me on my own.

Where assistance or advice was received or where I used resource material form a workbook, policy wording, internet or any other printed sources, this has been acknowledged and referenced. I further declare that I understand that plagiarism is a punishable offence as it constitutes the theft of another’s intellectual property rights.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Learner signature Date

Declaration by mentor

I (*mentor name and surname)* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, ID Nr \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_hereby declare that the learner is being mentored by myself and that the functions listed and the working hours is a true reflection of the learner’s situation. According to my knowledge I declare that this is his/her own work.

**Instructions to the Mentor**

Congratulations on your appointment as a mentor to the learner. This is a very responsible assignment because you have been tasked with the responsibility of rounding off the learner’s practical exposure.

You must ensure that you are familiar with all aspects of the work covered in this logbook because you must keep a daily account of the learner’s performance.

You are also required to report to the learnership coordinator skills development facilitator, or as agreed between yourself, the coordinator and the learner regarding the learner’s progress. Your responsibilities as mentor are as follows:

* Study the logbook and acquaint yourself with its content and format
* Remember this logbook is the learner’s full record of learning and workplace exposure. All activities which the learner participates in must be recorded, and all documents produced in relation to this learnership must form part of the record of learning
* Get all the learners together and explain its purpose to them and also what is required of them
* Remember the mentor is the creator of learning and exposure opportunities. You should therefore not confine the learner’s exposure to this logbook alone
* Continuously guide them in doing the assignments and arranging the planned exposure with the relevant departments
* File all duplicate records of learning on a file for each learner
* Send all original records to the training provider at the end of end of each month. The Seta also requires that copies be held at the companies
* This logbook need not be followed chronologically, but please note that the learner is required to work through the entire logbook by the end of the learning period

**Responsibilities of the Employer**

* Creating an atmosphere conducive to learning
* Giving learners ample access to the working environment. Remember that learners should be productive employees to get practical exposure to all aspects of the transport operation as required by the learnership
* Ensure that learners, mentors and assessors attend all training required and arrange and pay travelling and accommodation costs
* Ensure availability of sufficient mentors and workplace assessors
* Training Provider Responsibility
* Provide all practical learning material in electronic and/or hard copy, depending on the circumstances
* Provide training for mentors and workplace assessors if required and provide the learning material
* Visit employers to monitor progress and provide guidance and feedback.
* Provide an online and telephonic support system to all mentors and learners

**Purpose of the Practical Skill Module**

The focus of the learning in this module is on providing the learner with an opportunity to apply functionalities appropriately when programming with C++ The learner will be required to:

PM-02-PS01 : Declare variables in C++

PM-02-PS02 : Declare strings in C++

PM-02-PS03 : Use operators in C++

PM-02-PS04 : Use conditions

PM-02-PS05 : Use switch statements in C++

PM-02-PS06 : Declare and initialise arrays in C++

PM-02-PS07 : Use loops in C++

PM-02-PS08 : Create and declare enumerations and references in C ++

PM-02-PS09 : Use exception handling functions to find and mitigate exceptions

PM-02-PS10 : Create, initialise and delete dynamic arrays

PM-02-PS11 : Use pointers in C++

PM-02-PS12 : Declare a char (character)

PM-02-PS13 : File handling in C++

PM-02-PS14 : Use C++ Struct

PM-02-PS15 : Create a class and an object

PM-02-PS16 : Overload an Operator

PM-02-PS17 : Use std::list in C++

PM-02-PS18 : Write and call a function in C++

PM-02-PS19 : Use date and time in C++

PM-02-PS20 : Debug code

**SECTION A- PM-02-PS01 : Declare variables in C++**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA0101 Declare variables
* PA0102 Initialise variables
* PA0103 Name variables
* PA0104 Assign a value to variables

***Applied Knowledge***

* AK0101 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC0101 Variables declaration and assigning a value are completed

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA0101 Declare variables |  |  |  |
| PA0102 Initialise variables |  |  |  |
| PA0103 Name variables |  |  |  |
| PA0104 Assign a value to variables |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION B- PM-02-PS02 : Declare strings in C++**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA0201 Declare a C-style character string
* PA0202 Declare a string class type
* PA0203 Access string values
* PA0204 Use string functions

***Applied Knowledge***

* AK0201 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC0201 Expected results with C++ strings are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA0201 Declare a C-style character string |  |  |  |
| PA0202 Declare a string class type |  |  |  |
| PA0203 Access string values |  |  |  |
| PA0204 Use string functions |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION C- PM-02-PS03 : Use operators in C++**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA0301 Use arithmetic operators to perform arithmetic/ mathematical operations on operands
* PA0302 Use relational operators for the comparison of the values of two operands
* PA0303 Use logical operators to combine two or more conditions/constraints or to complement the evaluation of the original condition in consideration
* PA0304 Use bitwise operators to perform bit-level operations on the operands
* PA0305 Use assignment operators to assign value to a variable
* PA0306 Use sizeof operator
* PA0307 Use comma operator
* PA0308 Use conditional operator
* PA0309 Transform the number of any system such as binary to decimal system

***Applied Knowledge***

* AK0301 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC0301 Expected results with C++ operators are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA0301 Use arithmetic operators to perform arithmetic/ mathematical operations on operands |  |  |  |
| PA0302 Use relational operators for the comparison of the values of two operands |  |  |  |
| PA0303 Use logical operators to combine two or more conditions/constraints or to complement the evaluation of the original condition in consideration |  |  |  |
| PA0304 Use bitwise operators to perform bit-level operations on the operands |  |  |  |
| PA0305 Use assignment operators to assign value to a variable |  |  |  |
| PA0306 Use sizeof operator |  |  |  |
| PA0307 Use comma operator |  |  |  |
| PA0308 Use conditional operator |  |  |  |
| PA0309 Transform the number of any system such as binary to decimal system |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION D- PM-02-PS04 : Use conditions**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA0401 We write a calculator that adds, subtracts, multiplies and divides numbers and is resistant to dividing by zero

***Applied Knowledge***

* AK0401 Functionalities of C++ programming language

**Scenario: Building a Library Book Management System in C++**

Data Structures and Classes:

Students start by defining a class called "Book" to represent the properties of a book, such as title, author, ISBN, and availability. They'll use C++ classes and objects to encapsulate these attributes.

class Book {

public:

string title;

string author;

string isbn;

bool available;

};

Input and Output:

To interact with the system, students will use C++ I/O features. They can create a menu to add, view, and borrow books, using functions like cin and cout.

void displayMenu() {

// Display menu options using cout

}

// Read user input using cin

int choice;

cin >> choice;

Arrays and Collections:

Students will use C++ arrays or other collection types to store and manage the list of books in the library.

const int MAX\_BOOKS = 100;

Book libraryBooks[MAX\_BOOKS];

Functions and Control Flow:

C++ functions allow students to modularize their code. They can create functions for adding, viewing, and borrowing books, which will help in structuring the program and implementing control flow.

void add Book(Book newsbook) {

// Add a book to the libraryBooks array

}

Object-Oriented Programming:

C++ supports object-oriented programming, allowing students to use inheritance, polymorphism, and encapsulation. For instance, they can create derived classes for different types of books like "Fiction Book" and "NonFictionBook."

Pointers and Memory Management:

C++ provides features like pointers. Students can use them to dynamically allocate memory for new books and manage the memory efficiently.

Book\* newBook = new Book;

File Handling:

Students can use C++ to read and write book data to files, allowing for persistent storage of the library's catalog.

Error Handling:

They can also implement error handling using try-catch blocks to handle exceptions in case of invalid user inputs or file I/O errors.

Templates and Standard Library:

Students can use C++ templates to create generic functions for sorting or searching books and leverage the C++ Standard Library for various data structures and algorithms.

By developing this library book management system, students will get hands-on experience with many essential C++ functionalities, providing them with a well-rounded understanding of the language. This scenario covers data structures, I/O, functions, control flow, OOP, memory management, and more, making it a comprehensive educational experience for C++ programming.

***Internal Assessment Criteria***

* IAC0401 Expected results with C++ conditional statements are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA0401 We write a calculator that adds, subtracts, multiplies and divides numbers and is resistant to dividing by zero |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION E- PM-02-PS05 : Use switch statements in C++**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA0501 Write code for switch statements to compare the value of a variable against a set of other values

***Applied Knowledge***

* AK0501 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC0501 Expected results with C++ switch statements are achieved

scenario for students to understand how to use C++ switch statements effectively. Let's imagine a simple program where students have to choose a day of the week and the program will provide information about that day, including whether it's a weekday or a weekend.

#include <iostream>

using namespace std;

int main() {

int day;

cout << "Enter a number (1-7) to represent a day of the week: ";

cin >> day;

switch (day) {

case 1:

cout << "Monday - It's a weekday." << endl;

break;

case 2:

cout << "Tuesday - It's a weekday." << endl;

break;

case 3:

cout << "Wednesday - It's a weekday." << endl;

break;

case 4:

cout << "Thursday - It's a weekday." << endl;

break;

case 5:

cout << "Friday - It's a weekday." << endl;

break;

case 6:

cout << "Saturday - It's the weekend!" << endl;

break;

case 7:

cout << "Sunday - It's the weekend!" << endl;

break;

default:

cout << "Invalid input. Please enter a number between 1 and 7." << endl;

break;

}

return 0;

}

In this scenario, students are asked to input a number between 1 and 7, representing a day of the week. The program then uses a switch statement to determine which day was selected and provides information about whether it's a weekday or a weekend. This scenario helps students understand how to use switch statements to make decisions based on different input values and how to handle unexpected input using the default case. It's a practical and simple example to introduce them to switch statements in C++.

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA0501 Write code for switch statements to compare the value of a variable against a set of other values |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION F- PM-02-PS06 : Declare and initialise arrays in C++**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA0601 Declare an array in C++
* PA0602 Initialise an array in C++
* PA0603 Use a pointer to an array
* PA0604 Access the elements of an array using their respective indexes
* PA0605 Access the values of an array
* PA0606 Add elements to an array
* PA0607 print array elements
* PA0608 Delete elements from an array
* PA0609 Find minimum and maximum values in an array
* PA0610 Copy an array
* PA0611 Sort arrays
* PA0612 Looping through arrays
* PA0613 Check if arrays are equal
* PA0614 Search arrays
* PA0615 Fill arrays

***Applied Knowledge***

* AK0601 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC0601 Expected results with C++ arrays are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA0601 Declare an array in C++ |  |  |  |
| PA0602 Initialise an array in C++ |  |  |  |
| PA0603 Use a pointer to an array |  |  |  |
| PA0604 Access the elements of an array using their respective indexes |  |  |  |
| PA0605 Access the values of an array |  |  |  |
| PA0606 Add elements to an array |  |  |  |
| PA0607 print array elements |  |  |  |
| PA0608 Delete elements from an array |  |  |  |
| PA0609 Find minimum and maximum values in an array |  |  |  |
| PA0610 Copy an array |  |  |  |
| PA0611 Sort arrays |  |  |  |
| PA0612 Looping through arrays |  |  |  |
| PA0613 Check if arrays are equal |  |  |  |
| PA0614 Search arrays |  |  |  |
| PA0615 Fill arrays |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION G- PM-02-PS07 :Use loops in C++**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA0701 Declare and initialise loop control variables
* PA0702 Execute the loop body
* PA0703 Declare and initialise do-while loop
* PA0704 Create nested do-while loop

***Applied Knowledge***

* AK0701 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC0701 Expected results with C++ For Loop are achieved
* IAC0702 Expected results with C++ Do-While Loop are achieved
* IAC0703 Expected results with C++ nested Do-While Loop are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA0701 Declare and initialise loop control variables |  |  |  |
| PA0702 Execute the loop body |  |  |  |
| PA0703 Declare and initialise do-while loop |  |  |  |
| PA0704 Create nested do-while loop |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION H- PM-02-PS08 : Create and declare enumerations and references in C ++ *Scope***

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA0801 Define values and declare an enumerated type
* PA0802 Create reference variables
* PA0803 Write a function that sPAps two values
* PA0804 Return a reference type variable

***Applied Knowledge***

* AK0801 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC0801 Expected results with C++ references and enumerations are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA0801 Define values and declare an enumerated type |  |  |  |
| PA0802 Create reference variables |  |  |  |
| PA0803 Write a function that sPAps two values |  |  |  |
| PA0804 Return a reference type variable |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION I- PM-02-PS09 : Use exception handling functions to find and mitigate exceptions**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA0901 Throw exceptions
* PA0902 Catch exceptions
* PA0903 Use a combination of the try and catch keywords

***Applied Knowledge***

* AK0901 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC0901 Expected results with exception handling in C++ are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA0901 Throw exceptions |  |  |  |
| PA0902 Catch exceptions |  |  |  |
| PA0903 Use a combination of the try and catch keywords |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION J- PM-02-PS10 : Create, initialise and delete dynamic arrays**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA1001Create a dynamic array using the new keyword
* PA1002Initialise dynamically allocated arrays
* PA1003Delete a dynamic array from the computer memory

***Applied Knowledge***

* AK1001 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC1001 Expected results with dynamic arrays in C++ are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA1001Create a dynamic array using the new keyword |  |  |  |
| PA1002Initialise dynamically allocated arrays |  |  |  |
| PA1003Delete a dynamic array from the computer memory |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION K- PM-02-PS11 : Use pointers in C++**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA1101 Declare pointers
* PA1102 Initialise pointers
* PA1103 Obtain the address of a variable
* PA1104 Access variables by its address using pointers

***Applied Knowledge***

* AK1101 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC1101 Expected results with pointers in C++ are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA1101 Declare pointers |  |  |  |
| PA1102 Initialise pointers |  |  |  |
| PA1103 Obtain the address of a variable |  |  |  |
| PA1104 Access variables by its address using pointers |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION L- PM-02-PS12 : Declare a char (character)**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA1201 Declare a char
* PA1202 Print ASCII values
* PA1203 Print char value
* PA1204 Input char
* PA1205 Convert char to string

***Applied Knowledge***

* AK1201 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC1201 Expected results with Char data types in C++ are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA1201 Declare a char |  |  |  |
| PA1202 Print ASCII values |  |  |  |
| PA1203 Print char value |  |  |  |
| PA1204 Input char |  |  |  |
| PA1205 Convert char to string |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION M- PM-02-PS13 : File handling in C++**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA1301 Open files
* PA1302 Close files
* PA1303 Write to files
* PA1304 Read from files
* PA1305 Extract characters from files
* PA1306 Compare content of two files
* PA1307 Load characters from cin stream directly to file
* PA1308 Peek characters without extracting
* PA1309 Return extracted character on stream
* PA1310 Write bytes to file
* PA1311 Count characters from last operation
* PA1312 Conduct error handling with IO stream

***Applied Knowledge***

* AK1301 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC1301 Expected results with file handling in C++ are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA1301 Open files |  |  |  |
| PA1302 Close files |  |  |  |
| PA1303 Write to files |  |  |  |
| PA1304 Read from files |  |  |  |
| PA1305 Extract characters from files |  |  |  |
| PA1306 Compare content of two files |  |  |  |
| PA1307 Load characters from cin stream directly to file |  |  |  |
| PA1308 Peek characters without extracting |  |  |  |
| PA1309 Return extracted character on stream |  |  |  |
| PA1310 Write bytes to file |  |  |  |
| PA1311 Count characters from last operation |  |  |  |
| PA1312 Conduct error handling with IO stream |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION N- PM-02-PS14 : Use C++ Struct**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA1401 Create a C++ structure
* PA1402 Create struct instances
* PA1403 Initialise a structure
* PA1404 Access a structure
* PA1405 Create a pointer to a structure
* PA1406 Pass a struct to a function as an argument
* PA1407 Combine data items with struct

***Applied Knowledge***

* AK1401 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC1401 Expected results with C++ Struct are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA1401 Create a C++ structure |  |  |  |
| PA1402 Create struct instances |  |  |  |
| PA1403 Initialise a structure |  |  |  |
| PA1404 Access a structure |  |  |  |
| PA1405 Create a pointer to a structure |  |  |  |
| PA1406 Pass a struct to a function as an argument |  |  |  |
| PA1407 Combine data items with struct |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION O- PM-02-PS15 : Create a class and an object**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA1501 Declare a class
* PA1502 Declare class objects
* PA1503 Create multiple objects
* PA1504 Access class members
* PA1505 Define class member functions
* PA1506 Initialise objects using constructors
* PA1507 Destroy class objects with destructors

***Applied Knowledge***

* AK1501 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC1501 Expected results with class and object in C++ are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA1501 Declare a class |  |  |  |
| PA1502 Declare class objects |  |  |  |
| PA1503 Create multiple objects |  |  |  |
| PA1504 Access class members |  |  |  |
| PA1505 Define class member functions |  |  |  |
| PA1506 Initialise objects using constructors |  |  |  |
| PA1507 Destroy class objects with destructors |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION P- PM-02-PS16 : Overload an Operator**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA1601 Overload an operator

***Applied Knowledge***

* AK1601 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC1601 Expected results with operator overloading in C++ are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA1601 Overload an operator |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION Q- PM-02-PS17 : Use std::list in C++**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA1701Define a standard list
* PA1702 Insert into a list
* PA1703 Delete from a list
* PA1704 Insert and remove items from anywhere with std::list

***Applied Knowledge***

* AK1701 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC1701 Expected results with std::list in C++ are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA1701Define a standard list |  |  |  |
| PA1702 Insert into a list |  |  |  |
| PA1703 Delete from a list |  |  |  |
| PA1704 Insert and remove items from anywhere with std::list |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION R- PM-02-PS18 : Write and call a function in C++**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA1801 Invoke built-in functions
* PA1802 Invoke user-defined functions
* PA1803 Use the Function declaration/prototype to declare a function without a body
* PA1804 Call a function
* PA1805 Pass arguments

***Applied Knowledge***

* AK1801 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC1801 Expected results with functions in C++ are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA1801 Invoke built-in functions |  |  |  |
| PA1802 Invoke user-defined functions |  |  |  |
| PA1803 Use the Function declaration/prototype to declare a function without a body |  |  |  |
| PA1804 Call a function |  |  |  |
| PA1805 Pass arguments |  |  |  |
| PA1801 Invoke built-in functions |  |  |  |
| PA1802 Invoke user-defined functions |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION S- PM-02-PS19 : Use date and time in C++**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA1901 Compile date and time
* PA1902 Compare date and time
* PA1903 Date and time now
* PA1904 Date and time print
* PA1905 Date and time format

***Applied Knowledge***

* AK1901 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC1901 Expected results with date/time in C++ are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA1901 Compile date and time |  |  |  |
| PA1902 Compare date and time |  |  |  |
| PA1903 Date and time now |  |  |  |
| PA1904 Date and time print |  |  |  |
| PA1905 Date and time format |  |  |  |
| PA1901 Compile date and time |  |  |  |
| PA1902 Compare date and time |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

**SECTION T- PM-02-PS20 : Debug code**

Given an applicable instruction and access to a learning platform, the learner must be able to:

* PA2001 Use debugging tools for diagnosing and fixing bugs and errors
* PA2002 Identify incorrect code and analyse how a program "flows"
* PA2003 Apply debugging steps to debug effectively and efficiently

***Applied Knowledge***

* AK2001 Functionalities of C++ programming language

***Internal Assessment Criteria***

* IAC2001 Expected results with debugging in C++ are achieved

**EVALUATION**

|  |  |
| --- | --- |
| **NAME:** …………………………………….….  **SURNAME:** ……………………………….…  **COMPANY**……………………………….…..  **ID**……………………………………………... | **EVALUATION CHECKLIST**  DATE:  TIME: |

|  |  |  |  |
| --- | --- | --- | --- |
| **EVALUATION CRITERIONS** | **COMPETENT** | **NOT YET**  **COMPETENT** | **COMMENTS OR ACTION REQUIRED** |
|  | | | |
| **Ability of the learner to:** |  | | |
| PA2001 Use debugging tools for diagnosing and fixing bugs and errors |  |  |  |
| PA2002 Identify incorrect code and analyse how a program "flows" |  |  |  |
| PA2003 Apply debugging steps to debug effectively and efficiently |  |  |  |
| **GENERAL COMMENTS OF OBSERVER:** | | | |
| **Date \_\_\_\_\_\_\_\_\_** | **Time started\_\_\_\_\_\_\_\_\_** | | **Time completed\_\_\_\_\_\_\_** |
| **MENTOR/SUPERVISOR NAME**  **\_\_\_\_\_\_\_\_\_\_\_\_** | **MENTOR/SUPERVISOR SIGNATURE**  **\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **LEARNER**  **(SIGNATURE)**  **\_\_\_\_\_\_\_\_\_\_\_\_** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LOGBOOK** | | | | |
| **Date** | **Assignment task** | **Start** | **Finish** | **Total Hours** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **Date** | **Learner signature** | **Date** | **Mentor/supervisor signature** | |
|  |  |  |  | |