higher education & training

Department:
Higher Education and Training
REPUBLIC OF SOUTH AFRICA

**NATIONAL CERTIFICATE (VOCATIONAL)**

# INTRODUCTION TO TECHNICAL PROGRAMMING

(Second paper)
**NQF LEVEL 3**

(10041063)

**5 December 2024 (X-paper)**
**09:00–12:30**

**This question paper consists of 13 pages.**

613Q2N2405

Copyright reserved

# DEPARTMENT OF HIGHER EDUCATION AND TRAINING
# REPUBLIC OF SOUTH AFRICA

NATIONAL CERTIFICATE (VOCATIONAL)
INTRODUCTION TO TECHNICAL PROGRAMMING
NQF LEVEL3
(Second paper)
TIME: 3½ HOURS
MARKS: 100

---

**INSTRUCTIONS AND INFORMATION**

1. Answer all the questions.

2. Read all the questions carefully.

3. Number the answers according to the numbering system used in this question paper.

**INSTRUCTIONS TO STUDENTS AND LECTURERS**

1. Students MUST be seated in the computer laboratory 30 minutes before the start of the examination to ensure that all computers have the appropriate programs.

2. These questions must be completed on a computer by making use of:
   - MSWord
   - mBlock/Scratch 3
   - Any applicable C/C++ IDE

   **NO HANDWRITTEN code or HAND DRAWN diagrams will be marked!**

3. Students must make sure that they print their work immediately after completing a question. Ensure that your EXAMINATION NUMBER appears on EACH printout. NO questions without a printed EXAMINATION NUMBER will be marked.

4. In the event of problems such as a power failure, computer breakdown or a printer breakdown, the invigilator will make the necessary arrangements to continue with the examination process once the problem has been resolved. Students will not be penalised with time lost under these circumstances.

5. All work on the computer should be saved at regular intervals to prevent loss of work. NO additional time may be allowed for work lost due to lack of saving.

6. NO student may print his/her work for another student or make a memory stick available to another student or access another student's work on the network. Any attempt to access any information from or transfer information to another student in whatever manner is a contravention of the examination rules and regulations and will be viewed in an extremely serious light.

Copyright reserved 

7. NO access to the INTERNET is allowed.

8. ALL PRINTOUTS or screen prints done during the session MUST be handed in.

9. Make sure that your examination number appears on all printouts.

10. Students may NOT copy the source code from the C/C++ IDE code editor to MS Word and then print it from the word processor.

11. Steps to print a user interface or any other screenshot:

    11.1 Run the solution.

    11.2 Hold down the ALT key and press the print screen (Print Scrn) key.

    11.3 Open a blank MS Word document and paste the screenshot.

    11.4 Add your EXAMINATION NUMBER against the left margin.

    11.5 Make a PRINTOUT of the page.

---

Copyright reserved 

## QUESTION 1

Study the program scenario and use the following guidelines to create a program.

**Program scenario**



In June 2024, a month before the Olympic Games in Paris, the city experienced one of the heaviest periods of rainfall recorded in 2024. It rained for five days nonstop. The data collected was measured in millimetres. The Olympics committee has chosen your company to record the amount of rain (in millimetres per day) and to perform calculations on the data. You have been appointed as the head of the project.

Follow the instructions below to create the mBlock/Scratch application.

1.1 Create a mBlock/Scratch application and choose your own Stage and Sprite. (1)

1.2 Create a list called Rainfall to keep a record of the rainfall for the 5 days. (1)

1.3 Include and initialise all appropriate variables. (2)

1.4 Prompt the user to enter rainfall for each of the five days. The values must be stored in the list. (2)

1.5 Determine and display the average rainfall. (3)

1.6 Determine and display how many days had above-average rainfall. (3)

1.7 Print out your code:
Make a screenshot of your code and paste it into an MS Word document.
Include the heading Question 1 and your Examination Number. (1)

1.8 Test your program with the following values:

- 255
- 375
- 200
- 300
- 350

(2)

Make a screenshot of your output and paste it into your MS Word document.
Print out your document.

SAVE your program as *Question1_EXAMINATION_NUMBER* using your examination number.

**[15]**

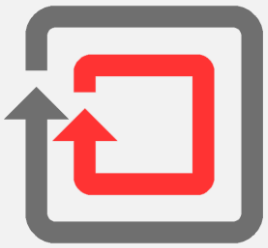
Copyright reserved 

## QUESTION 2

Study the program scenario and create a C++ program using the following guidelines.

**Program scenario**

You must create a C++ program that will prompt the user for the number of rows to print. Your program must then display the pattern in the sample output screens below.



You must implement nested loops as part of the solution.

```
Please enter the number of rows
7

7#6#5#4#3#2#1#
6#5#4#3#2#1#
5#4#3#2#1#
4#3#2#1#
3#2#1#
2#1#
1#
```

```
Please enter the number of rows
4

4#3#2#1#
3#2#1#
2#1#
1#
```

Follow the instructions below to create the C++ application.

2.1 Add your Examination Number as a comment in the code. (1)

2.2 Prompt the user for the number of rows and store the value in an appropriate variable. (3)

2.3 Use the input from the user to generate the pattern of numbers by using appropriate nested loops. (9)
TIP: Only a single character, a hash (#, and a newline character) may be printed at a time where appropriate.

Print your program code from the IDE.

Make a screenshot of your output and paste it into an MS Word document. Include the heading Question 2 and your Examination Number. Print out your document.

SAVE your program as: Question2_EXAMINATION_NUMBER.cpp using your EXAMINATION NUMBER.

**[13]**

## QUESTION 3

Study the program scenario and create a C++ program using the following guidelines.

**Program scenario**




Encrypting messages has always been popular, especially in the military. Encrypting a message involves changing it so that it is difficult for other people to read it unless you have the cypher, which acts as a key to unlock the encrypted messages. The use of encryption goes back to ancient times. The easiest method of encrypting plain text is the reverse cypher, which involves printing the message in reverse letter order.

**Example 1:**

Original message: ONE
Reverse order: ENO

**Example 2:**

Original message: `egaugnal gnimmargorp lufrewop a si ++C`
Reverse order: `C++ is a powerful programming language`

You must create a C++ program that prompts the user for a message to encrypt using the reverse cypher. The program must then display the message in reverse letter order.

The FIGURE 3.1 below demonstrates what the user interface looks like with sample input and output:

```
Message encryption program

Enter a message to encrypt: This is the original message.

Original message: This is the original message.
Reversed (encrypted) message: .egassem lanigiro eht si sihT
```

```
Message encryption program

Enter a message to encrypt: egaugnal gnimmargorp lufrewop a si ++C

Original message: egaugnal gnimmargorp lufrewop a si ++C
Reversed (encrypted) message: C++ is a powerful programming language
```

**FIGURE 3.1 – Sample inputs and outputs**

Follow the instructions below to create the C++ program.

3.1 Include the header file needed to use string functions. (1)

Copyright reserved 

3.2 Declare a void function prototype called `reverseIT()`. (1)

3.3 Declare TWO global string variables named `sOriginal` and `sReverse`, and initialise them both as empty strings. (2)

3.4 Include the following in the main function:

3.4.1 Display a program heading **`Message Encryption program`** (1)

3.4.2 Prompt the user to enter the original string to encrypt and store the string in the variable `sOriginal;` (2)

3.4.3 Call the function **reverseIT()** to encrypt the message. (1)

3.4.4 Display the original and the encrypted string stored in the `sReverse` variable;
(Refer to FIGURE 3.1 to guide you) (2)

3.5 Write the implementation code for the void function called `reverseIT()`.

The function should reverse the string in the `sOriginal` global variable and store the reversed string in the `sReverse` variable.

(7)

Test your program by using the following input string.

**Sample data:**

```
Message encryption program

Enter a message to encrypt: !ysae si gnidoC

Original message: !ysae si gnidoC
Reversed (encrypted) message: Coding is easy!
```

Make a screenshot of your output and paste it into an MS Word document. Include the heading Question 3 and your Examination Number. Print out your document.

Print your program code from the IDE.

**[17]**

## QUESTION 4

Study the program scenario and create a C++ program using the following guidelines.

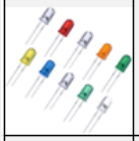
**Program scenario**

Ramsammy's Electro Market sells LEDs and Resistors for TVET students in a special sale.




Students can buy resistors in packs of 10 or LEDs in packs of 10.

The following table presents the rates and the prices if packs are bought in larger quantities.

| | Code | Product | Unit Price | More than 100 | More than 500 |
|---|---|---|---|---|---|
|  | A | LEDs 10 Pack | 2.85 | 2.28 | 1.9 |
|  | B | Resistors 10 Pack | 0.95 | 0.76 | 0.54 |

Create a complete C++ program that calculates and displays the selected product, quantity and the total price due for the selected component.

- 1 to 99 Packs with 10 LEDs cost R2.85 per pack; more than 100, but less than 501 costs R2.28 per pack and more than 500 packs cost R1.90 per pack
- 1 to 99 Packs with 10 Resistors cost R0.95 per pack; more than 100, but less than 501 costs R0.76 per pack and more than 500 packs cost R0.54 per pack

Include all the necessary header files.

Follow the instructions below to create the C++ program.

4.1 Create a void function named **`displayPriceList()`** to display the following information on the screen. (3)

```
Product Pricing Information:
Code | Product       | Unit Price (Rands) | >100   | >500
-----|---------------|--------------------|--------|------
A    | LEDs 10 Pack  | 2.85               | 2.28   | 1.9
B    | Resistors x10 | 0.95               | 0.76   | 0.54
```

4.2 Declare and initialise appropriate global variables. (1)

Copyright reserved 

4.3 Create a value-returning function named **`calculatePrice()`** that will return the correct total price based on the product code and the number of packs required. Also, save the description of the product based on the product code. (13)

4.4 Include the following in the main() function:

4.4.1 Call the **`displayPriceList()`** function. (1)

4.4.2 Read in the appropriate data. (8)

- The user should be allowed to enter an **`a`** or **`A`** as a product code as well as a **`b`** or a **`B`**
- The user may only enter values from 1 to 1000 for the quantity. If this is not the case, an error should be shown, and the user should be prompted to enter the quantity again until a valid quantity is entered. (**Hint:** Use a while loop)

4.4.3 Call the function **`calculatePrice()`** and display the total price due for the product selected. (7)

The name of the selected product and quantity should also be displayed.

Ensure that the total price is indeed formatted with two decimals.

Examples of sample runs:

**Sample run 1**

```
Product Pricing Information:
Code | Product       | Unit Price (Rands) | >100   | >500
-----|---------------|--------------------|--------|------
A    | LEDs 10 Pack  | 2.85               | 2.28   | 1.9
B    | Resistors x10 | 0.95               | 0.76   | 0.54

Enter the product code (A, B): a
Enter the quantity: 150
Total Price for 150 LEDs 10 Pack is R342.00
```

**Sample run 2**

```
Product Pricing Information:
Code | Product       | Unit Price (Rands) | >100   | >500
-----|---------------|--------------------|--------|------
A    | LEDs 10 Pack  | 2.85               | 2.28   | 1.9
B    | Resistors x10 | 0.95               | 0.76   | 0.54

Enter the product code (A, B): B
Enter the quantity: 1005
Invalid quantity.
Enter the quantity: 0
Invalid quantity.
Enter the quantity: 35
Total Price for 35 Resistors x10 is R33.25
```

Copyright reserved 

Make a screenshot of your output and paste it into an MS Word document. Include the heading Question 4 and your Examination Number. Print out your document.

Print your program code from the IDE.

**[33]**

**QUESTION 5**

Study the explanation of a C++ string function and the sample program and attempt the following question based on the problem statement.

**The to_string function**

# std::to_string in C++

Last Updated : 24 Sep, 2024

In C++, the **std::to_string** function is used to convert numerical values into the string. It is defined inside **<string>** header and provides a simple and convenient way to convert numbers of any type to strings.

In this article, we will learn how to use **std::to_string()** in C++.

## Syntax

```
std::to_string(val);
```

### Parameters

- **val:** It is the value which we have to convert into string. It can be of any numeric data type like integer, long long, double, float, long double.

### Return Value

- A string containing the representation of **val** as a sequence of characters.

Source [https://www.geeksforgeeks.org/stdto_string-in-cpp/]

**Sample program**

```cpp
#include <iostream>
#include <string>
#include <conio.h>
using namespace std;

int main() {
    // Integer variables
    int Number1 = 23;
    int Number2 = 9;

    // Floating-point variable
    double floatNumber = 3.14159;

    // Converting integers to strings
    string str1 = to_string(Number1);
    string str2 = to_string(Number2);

    // Converting floating-point to string
    string strFloat = to_string(floatNumber);

    // Displaying the results
    cout << "Integer to string conversions:" << endl;
    cout << "Number1 (23) as string: " << str1 << endl;
    cout << "Number2 (9) as string: " << str2 << endl;
    cout << "str1 = " << str1 << endl;
    cout << "str2 = " << str2 << endl;
    cout << "Length of str1: " << str1.length() << endl;
    cout << "Length of str2: " << str2.length() << endl;

    cout << "\nFloat to string conversion:" << endl;
    cout << "floatNumber (3.14159) as string: " << strFloat << endl;
    cout << "Length of strFloat: " << strFloat.length() << endl;

    // Demonstrating string concatenation
    string combined = str1 + str2;
    cout << "\nConcatenated string: str1 + str2 = " << combined << endl;

    getch();
    return 0;
}
```

**Output**

```
Integer to string conversions:
Number1 (23) as string: 23
Number2 (9) as string: 9
str1 = 23
str2 = 9
Length of str1: 2
Length of str2: 1

Float to string conversion:
floatNumber (3.14159) as string: 3.141590
Length of strFloat: 8

Concatenated string: str1 + str2 = 239
```

Copyright reserved 

Using your new knowledge of the `to_string()` function. Read the following information snippet and write a C++ program as described after that.

**Gauteng Number plates**




A typical Gauteng number plate consists of:

- two random alphabetical consonant characters followed by
- a space followed by
- a numerical value (00 to 99). It must be a randomly generated two-digit number. If the random number is a single digit, insert a 0 before the digit to make it a two-digit number. After that, it is followed by
- a space followed by
- two additional random alphabetical consonant characters followed by
- GP

Note: The consonants are all the alphabet letters, excluding A, E, I, O and U.

Follow the instructions below to create the C++ program:

5.1 Create a C++ program with the necessary headers and define and initialise the following global character array named consonants: (5)

```
char consonants[21] = {'B', 'C', 'D', 'F', 'G', 'H', 'J', 'K',
'L', 'M', 'N', 'P', 'Q', 'R', 'S', 'T', 'V', 'W', 'X', 'Y',
'Z'};
```

Copyright reserved 

5.2 Add the code for the **`main()`** function, which will generate and print 10 randomly created GP number plates based on the rules described above. Generate the parts of the numberplate by using concatenation in a string variable named **`plate`** before displaying the randomly generated GP plate. (17)

**Sample run 1**

```
10 Generated Number Plates:

XV 05 WJ GP
NM 30 HG GP
TQ 71 SK GP
PY 48 WS GP
NQ 63 JF GP
CN 25 MJ GP
GL 25 JG GP
FX 85 CD GP
PY 58 YV GP
VH 19 TB GP
```

Make a screenshot of your output and paste it in MS Word. Include the heading Question 5 and your EXAMINATION NUMBER. Print out your document.

SAVE your program as: Question5_EXAMINATION_NUMBER.cpp using your EXAMINATION NUMBER.

*Ensure that you added your examination number as a comment to the first line of the program.*

Print your program code from the IDE.

**[22]**

**TOTAL: 100**

Copyright reserved