**Examination Answer Book**

**UNIVERSITY EXAMS**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| REGISTRATION NUMBER | | | | | | | | | VU-BIT-2403-0080-EVE | | | | | | |
| Title of The Program (eg BBA, BSC, BPH, BSWA) | | | | | | | | | | | | | BIT | | |
| Bachelor of Information Technology | | | | | | | | | | | | | | | |
| Department | | | | Other Depts in Faculty of Science and Technology | | | | | | | | | | | |
| Faculty | Faculty of Science and Technology | | | | | | | | | | | | | | |
| Year Of study (YrI , YrII, YrIII, or YrIV) | | | | | | | | | | | 1 | | | | |
| Module Code and Name | | | | | | | 1203 ST | | | | | | | | |
| Programming Fundamentals | | | | | | | | | | | | | | | |
| Semester | | | 2 | | | | | | | | | | | | |
| (Copy from the heading to the Examination Paper) | | | | | | | | | | | | | | | |
| Retake: | | Yes | | |  | | | No | |  | | (Tick whichever is applicable) | | | |
| Date of examination | | | | | | Sun Sep 22 2024 08:00:00 GMT+0300 (East Africa Time) | | | | | | | | | |
|  | | | | | |  | | | | | | | | | |
| **DIRECTIONS TO CANDIDATES (Turn to page ii for more instructions).** | | | | | | | | | | | | | **FOR USE BY EXAMINERS ONLY** | | |
| **Question Number** | **Internal Examiner** | **External Examiner** |
| 1. Leave margin blank. 2. Begin each answer on a fresh page. 3. Write the number of each question and theCandidate's Number at the top of each page. 4. Write the numbers of the questionswhich you have attempted, with subsections where necessary, in the spacesprovided below | | | | | | | | | | | | |
| 1 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **TOTAL** |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NUMBER OF QUESTIONS** you have answered in the order in which you have written them | | | | | | | | |
|  |  |  |  |  |  |  |  |  |

**How and where should I submit my examination script?**

Every student will be required to attend their examination via [VClass Students Portal](https://vclass.ac/) E.g. you go to [www.vclass.ac](http://www.vclass.ac) and login, to your account, then on the left sidebar menu **click on Examinations**.

Under examinations you will see the following: -

1. Instructions for that particular examination with time required to finish your examination as per instructions,
2. A student will be required to download the question paper and the answer sheet provided by the university within the same module examination, or a student can be required to attempt structured questions within the system depending on how the examination was set.
3. Submission of answered questions is done,
4. Student is required to click to **consent** to show that the answered exam belongs to them.
5. **Note** that if an examination is for download, a student will be required to download the question paper and answer sheet, write their examination within the given stipulated time.
6. Required to scan and upload back the answered booklet through the same portal as per format available.
7. Examinations uploaded will directly be received by the Registry department.
8. Students here are required to use [VClass e-Learning system](https://vclass.ac)for all examinations and for any failure they can contact the Registry department for guidance.
9. No late submission will be accepted.

**Avoid any examination malpractice because this will attract severe penalties such as invalidating the exams answered script whose consequences will attract retakes.**

Qn 5.

a)

i. A data type is the type of data a variable can hold. For example, a Boolean variable can have Boolean data, and an integer variable can hold integer data.

For example; int score = 99;

score here is an int datatype variable. The variable score can only store 2-byte or 4-byte integers, depending on the compiler/system.

In C++, data types can be classified as follows:

1. Primitive/Built-in Datatypes

2. Derived Datatypes

3. Abstract/User-defined Datatypes

ii. Variables is a name given to a memory location. It is the basic unit of storage in a program. The value stored in a variable can be changed during program execution. A variable is only a name given to a memory location, all the operations done on the variable effects that memory location.

In C++, there are different types of variables, for example:

 int - stores integers (whole numbers), without decimals, such as 123 or -123

 double - stores floating point numbers, with decimals, such as 19.99 or -19.99

 char - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes

 string - stores text, such as "Hello World". String values are surrounded by double quotes

 bool - stores values with two states: true or false

iii. Algorithm is the logic or plan for solving a problem represented as a simple step-by-step description. An algorithm represents the thinking process for solving a problem in an abstract yet precise way, rather than the answer itself.

For example, an algorithm that calculates the square of a given number.

• Input: the input data is a single-digit number (e.g., 5).

• Transformation/processing: the algorithm takes the input (number 5) and performs the specific operation (i.e., multiplies the number by itself).

• Output: the result of the calculation is the square of the input number, which, in this case, would be 25 (since 5 \* 5 = 25).

iv. Function is a block of code that performs some operation. It can optionally define input parameters that enable callers to pass arguments into the function. A function can optionally return a value as output. It is useful for encapsulating common operations in a single reusable block, ideally with a name that clearly describes what the function does.

For example. The following function accepts two integers from a caller and returns their sum; a and b are parameters of type int as seen below.

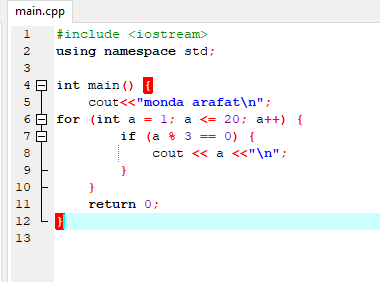
int sum (int a, int b)

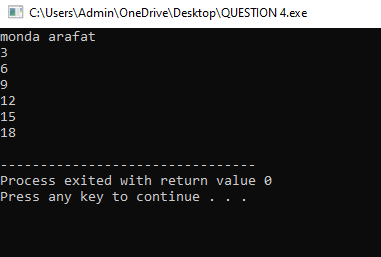
{

return a + b;

}

b)





c) Comments are texts written in a program along with the code to provide descriptions and key points about what’s happening in the code. These comments are mostly supported in all programming languages to make the code readable. Comments in C++ are mostly of two types- single-line comments and multi-line comments.

Single Line Comments

A single-line comment is a comment that consists of one line only. That is, it can occupy only a single line of the code. These comments are represented by two forward slashes (//). Anything written after // until the end of the first line is a single-line comment. These comments are only for the programmers to understand the code better. The compiler ignores everything written after //, and hence the comments do not affect the program directly in any way.

Syntax of Single line comment

// declaring a string s

string s;

// initializing the string 's' as “unstop”

s= “unstop”;

Use single-line comments:

To explain variable declarations or function parameters.

To clarify complex logical operations.

To provide a quick note about a specific line of code.

Multi-Line Comment

C++ also supports multi-line comments, which are spread over many lines to provide a description of the code. That is, these comments can span multiple lines in a code. They are enclosed in the delimiters /...../. That is, anything written between /\* until the end of the / is a multi-line comment. Just like single-line comments, these comments are also only for the programmers to understand the code better. And again, the compiler ignores everything written between /..\*/, and hence the comments do not affect the program directly in any way.

Syntax of multi-line comment

/\*

C++

Multi

Line

Comment

\*/

Use multi-line comments:

To document functions or classes.

To explain complex algorithms or logic.

To provide copyright or licensing information.

QUESTION.3 A

I. An array:

An array is a fundamental data structure that stores a collection of elements of the same data type in the memory locations.

So an array can be deployed in following way;

Declaring and Initializing Array

ie

 Static initialization: Specify values at declaration time.

int scores[5] = {90, 80, 70, 60, 50};

 Dynamic initialization: Initialize after declaration.

int scores[5];

scores[0] = 90;

scores[1] = 80;

Using Arrays

1. Indexing: Access elements using their index.

int value = scores[0]; // value = 90

2. Loops: Iterate through elements.

Using namespace std;

for (int i = 0; i < 5; i++) {

cout << scores[i] <<endl;

}.

Question. 3.A

II. Multi-Dimensional Arrays

Multi-Dimensional Arrays (MDAs) are data structures that store elements in a tabular or matrix format, allowing for efficient storage and manipulation of data with multiple dimensions.

1. Two-Dimensional (2D):

So here were looking at matrix i.e row and column

int matrix[3][4] = {{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}};

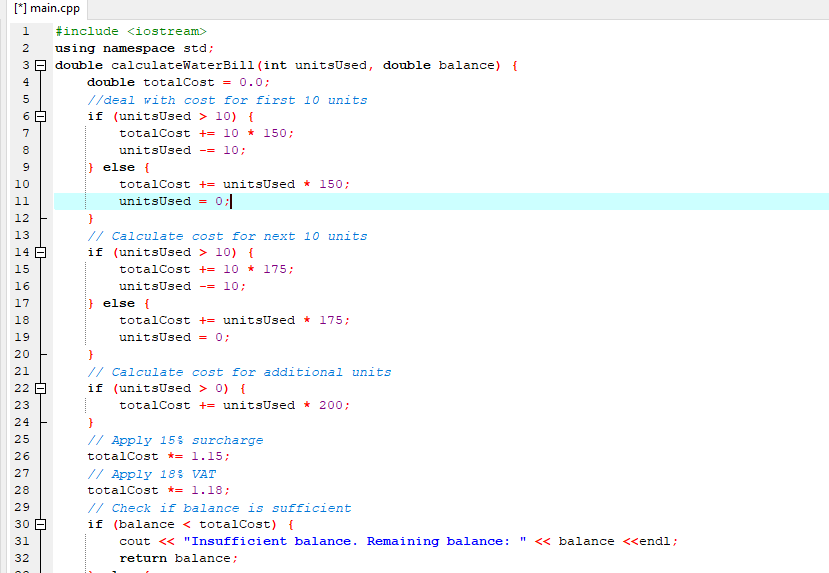
2. Three-Dimensional (3D):

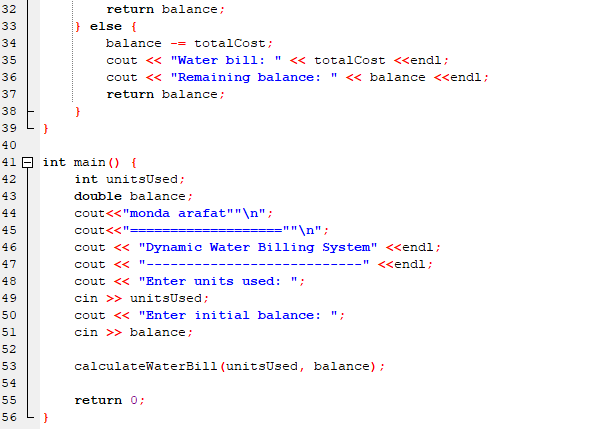
int cube[2][3][4] = {{{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}},

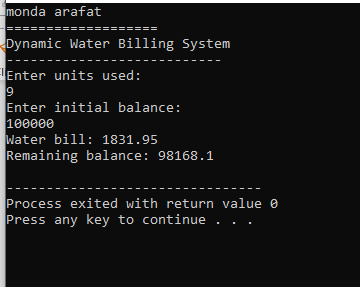
{{13, 14, 15, 16}, {17, 18, 19, 20}, {21, 22, 23, 24}}};

So above were showing the dimension the matrix

B,

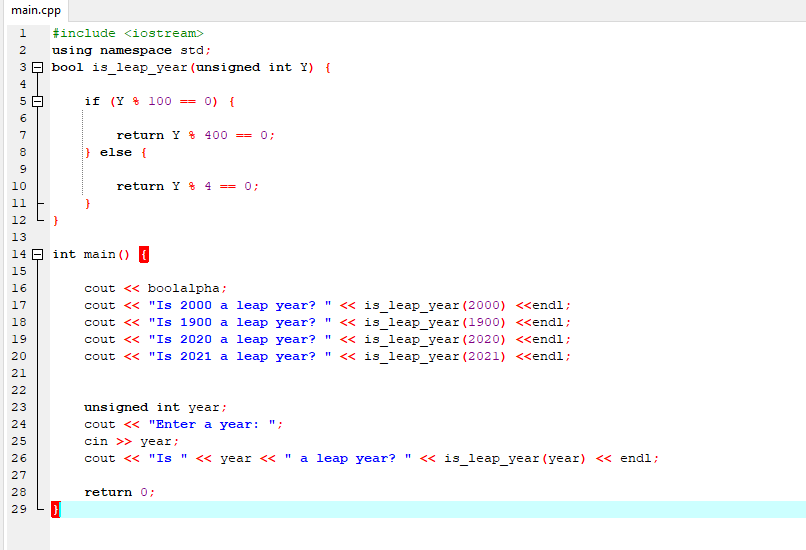


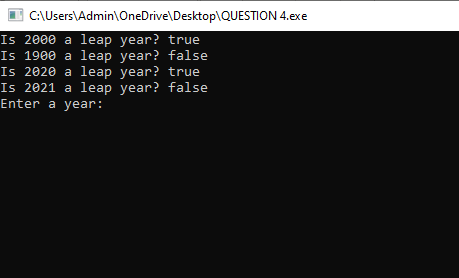




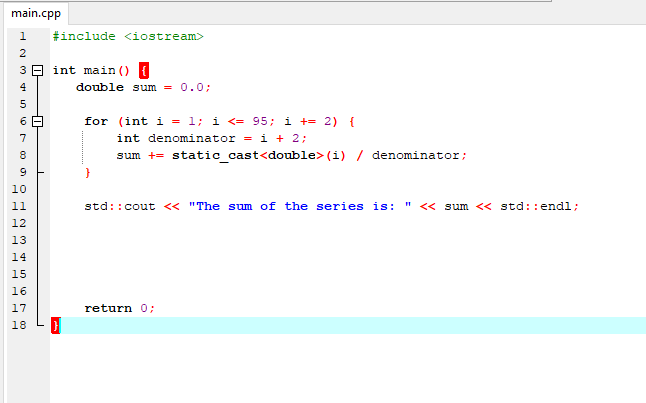
QUESTION 1.

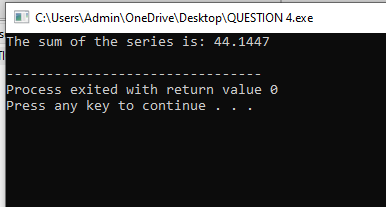
(A)





1(B)





QUESTION 4 (A)

**1. Syntax Errors**

* **Definition**: These occur when the code violates the grammatical rules of the programming language. They arise from incorrectly written statements, such as missing punctuation or improperly structured commands.
* **Example**: In Python, forgetting a colon after an if statement:

python

Copy code

if x > 10

print("x is greater than 10") # SyntaxError: Expected colon after 'if' statement

* **Impact**: Syntax errors prevent the code from being executed. The program won't run until all syntax errors are resolved, making them the first errors a programmer should fix.

**2. Runtime Errors**

* **Definition**: These occur while the program is running and typically happen when the program encounters an operation that it cannot execute, such as dividing by zero or accessing an out-of-bound index in an array.
* **Example**: Dividing a number by zero in Python:

python

Copy code

result = 10 / 0 # ZeroDivisionError: division by zero

* **Impact**: Runtime errors cause the program to crash during execution, and the programmer must handle them using exception handling techniques to ensure the program can fail gracefully.

**3. Logical Errors**

* **Definition**: These occur when the code executes without errors, but the output is not what was expected due to incorrect logic in the code. These are the hardest to detect as the program does not crash, but it does not behave as intended.
* **Example**: A function meant to calculate the average of a list but sums the values incorrectly:

python

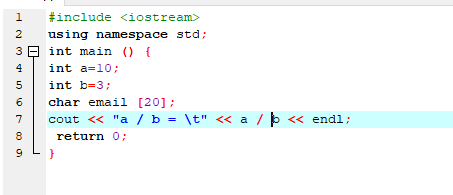
Copy code

def average(numbers):

return sum(numbers) \* len(numbers) # Logic Error: should divide by len(numbers)

* **Impact**: Logical errors can lead to incorrect results or faulty program behavior. They require careful debugging and testing to identify and fix.

4(b)



* 1. , The variable email is declared as an array of characters with a size of 20. This means it can store up to 19 characters plus a null terminator (\0) at the end to mark the end of the 19string.

(III)



(IV) python

Copy code

email = input("Please enter your email: ") # Get input from user

print("The email you entered is:", email) # Print the email to the screen

**Explanation:**

1. input("Please enter your email: "): The input() function prompts the user to enter data. The string inside the parentheses (the prompt) is displayed on the screen, and the user types their input, which is then stored in the variable email.
2. print("The email you entered is:", email): The print() function displays the content of the email variable, along with a message, on the screen.

QUESTION 4 (C)

