MY CHOICE
Date
Page No

I.1 Assignment

(1) Write a brief note on DSA along with its importance.

DSA or Data Structures and Algorithams, forms the fundamental building blocks of computer science and play a critical role in the development of efficient & scalable software solutions.

→ Data structures refers to the organization and storage of data in a computer's memory, while algorithams are step by step procedure or formulas for solving problems and performing.

→ Together they enable programmers to design and implement efficient solutions to complex computational problems.

Importance:

1. Optimized performance:
   Efficient algorithams are crucial for optimizing the performance of software. DSA enables the selection of appropriate data structure and algorithm to achive optimal time & space complexity

2. Problem solving

DSA provides a systematic approach to problem solving. It helps programmers understand the nature of problems & design efficient solutions.

3. Algorithmic Thinking

DSA enhances algorithmic thinking, enabling programmers to device logical and effective solutions. It encourages the development of a structured thought process when tackling computational problems.

4. Code Reusability

Proficiency in DSA facilitates the creation of modular and reusable code. Well-designed data structure and algorithms can be applied to different problems with minimal modification.

5. Interviews & Competition

It's a crucial aspect of technical interviews for software engineering positions. Many tech companies asses conditions based on their ability to solve algorithmic and data structure related programs.

6. Scalability

DSA plays a pivotal role in designing scalable systems, systems that are expected to handle large amount of data or requests next well-designed (optimized) algorithams and data structures to maintain performance.

2. What is Data staues? Explain with its types.

A data structure is a way of organizing & storing data in a computer so that it can be used efficiently.

→ It defines a set of operations that can be performed on the data, as well as the relationships between the data elements.

→ The choice of a particular data structure depends on the type of problem being solved and the operations that needs to be performed on the data.

Types:

- → Arrays
- → Linked Lists
- → Stacks
- → Ques
- → Trees
- → Graph
- → Hash Tables
- → Heaps

1.2 Assignment

(1) Explain the term "programming"
A set of instructions for computer to perform specific task and solve problems.

→ Programming impacts our world in countless ways:
- From the websites you visit,
- The apps you use.
- The games you play and the AI assistances you interact with.

Steps in programming:
1. Problem definition
2. Algorithm design
3. Coding
4. Compilation | Interpretation
5. Testing
6. Debugging
7. Documentation
8. Maintance

(2) Describe the history and importance of C++.

History:

→ C++ programming language was developed in 1979 by bjarne Stroustrup at bell laboratories of AT&T (American Telephone & Telegraph located in U.S.A.

→ Bjarne Stroustrup is known as the founder of C++ language.

→ It was develop for adding a feature of OOP (Object Oriented programming) in C language.

→ C++ programming is "relative" (called a superset) of C, it means any valid C program is also a valid C++ program.

Importance:

C++ is a powerful & versatile language with a crucial role in various invenations.

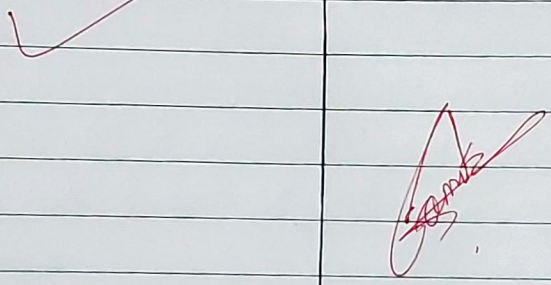
→ It's performance, versatility and established community make it a valuable skill for the programmers seeking to take a complex projects and pass the boundaries of software developments

1. Performance & efficiency
2. Versatility and natural purpose.
3. Object oriented programming (OOP) support
4. Large community
5. Legacy codebase & compatibility
6. low-level language.

(3) State the difference between compiler and interpreter.

Comp^{n}	Compiler	Interpreter
Input	It takes an entire programe at a time	It takes a single line of code at a time.
Output	It generates intermediate object code.	It does not produce any intermediate object code.
Working mechn^{m}	The compilation is done before execution	Compilation & execution takes place simulancously.
Speed	Comperatively faster	Slower
Memory	Memory requirement is more due to the creation OF object code.	It requires less memory as it does not create interme diate object code.

Errors	Display all errors after compilation, all at the same time.	Display error of each time one by one.
Error detectin	Difficult	Easier compaaratively.
Pertain progrn lang.	C, C++, C#, Scala, typescript uses compiler.	PHP, perl, Python, Ruby, uses an interpreter.



Date:
Page No.:

1.3 Assignment

(1) Explain data types, variables and keywords in detail with example.

* Data Types:

A data type is a keyword that defines what type of data a variable can hold. For example the data type int can hold whole numbers, while data type float can hold numbers with decimal points.

→ int: Whole numbers such as 1,456, 322
→ char: Contains characters, such as letters, numbers, spaces and symbols.
→ string: contains a set of character, such as "Hello", "How are you".
→ float: Contains numbers with a decimal point such as 9.1, 2.6552, 3.1425128
→ double: stores fractional numbers, such as 2.34 and 1.145
→ Boolean: Contains only one of two possible values, such as true or false.

* Variables:

A variable is a name associated with a memory location in the computer, where you can store a value that can change or vary

→ Declaration: Specifying the type and name of the variable.
→ int age;

→ Initialization: Assigning a value to the variable at the time of declaration
→ int age = 22;

* Keywords:
Keywords are reserved words in c++ that have special meaning to the compiler. They cannot be used as variable names.

- auto
- const
- bool
- explicit
- new
- struct
- throw
- bool
- union
- int
- virtual
- long
- void
- mutable
- while.
- continue
- friend
- operator
- switch
- true
- break
- double
- float
- catch
- public
- char
- register
- class
- return
- private
- short
- try
- case
- default
- for
- protected
- sizeof
- delete.
- static
- else
- this
- enum
- typedef

(2) Describe boolean data type in detail with example.

The boolean data type is a data type that store one of two values; true or false. It's named after George bool, who developed boolean algebra, a system of logic that uses true and false values.

→ In C++, the boolean data types is declared using the keyword bool. For example the following code declares a boolean variable name is_valid:

```
bool is_valid;
```

→ Boolean variables can be used in conditional statements to control the flow of program.

→ Boolean variables can also be used in mathematical expressions.

→ example:

```
include<iostream>
using namespace std;

int main(){
    bool is_valid = true;
```

```
    if (is_valid) {
        cout << "The value is valid." << endl;
    } else {
        cout << "The value is not valid!" << endl;
    }

    return 0;
}
```