

## **BIT 2203: DATA STRUCTURES AND ALGORITHMS**

### **Assignment 1**

**Using a programming language of your choice, in groups, write code to represent each of the data structure classification and types.**

**Research where the data structures types are applied and give reasons why.**

**Give examples of applications that are using the data structure type and algorithm. Give reasons why.**

**Research how data structures and algorithms work within systems.**

## **APPLICATIONS OF DATA STRUCTURES & ALGORITHMS IN SYSTEMS**

### **1. ARRAY**

#### **Where Arrays Are Applied**

Operating systems, databases, embedded systems, scientific computing, web applications.

#### **Example Applications**

Student management systems, image processing software, e-commerce platforms.

#### **Algorithms Used**

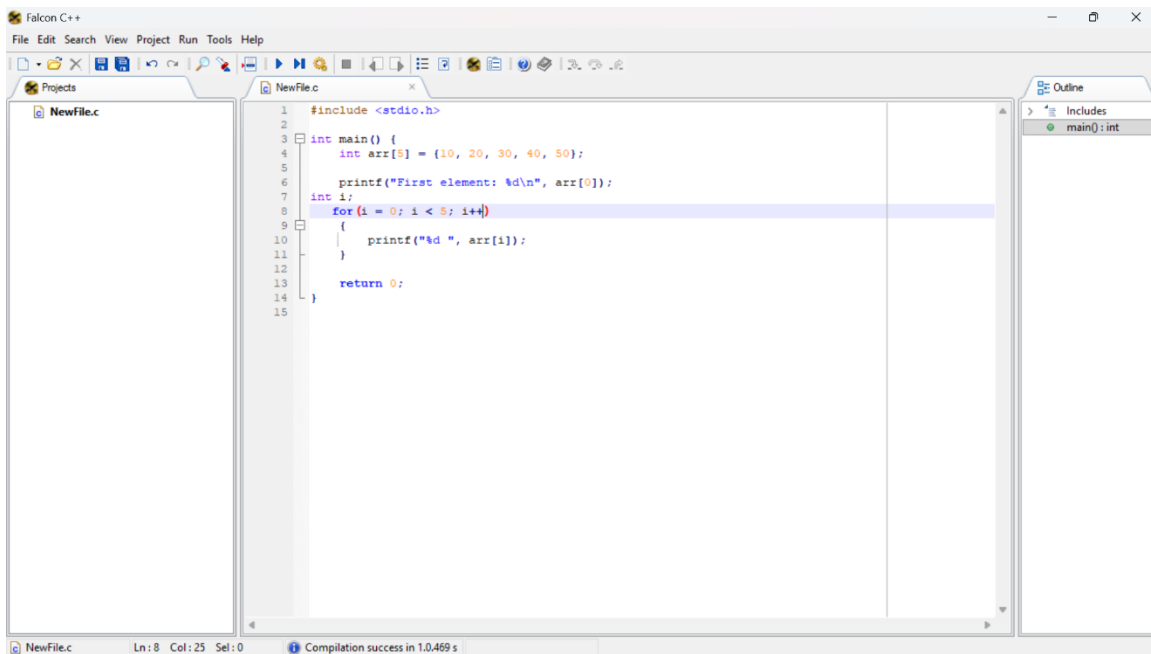
Linear Search, Binary Search, Bubble Sort, Quick Sort.

#### **Reasons for Using Arrays**

Fast access ( $O(1)$ ), simple structure, efficient for fixed-size data.

#### **How Arrays Work Within Systems**

Arrays use contiguous memory allocation, allowing fast access and high performance.



## 2. LINKED LIST

### Where Applied

Memory management, dynamic storage, system software, multimedia applications.

### Example Applications

Music playlists, undo/redo systems, file navigation.

### Algorithms Used

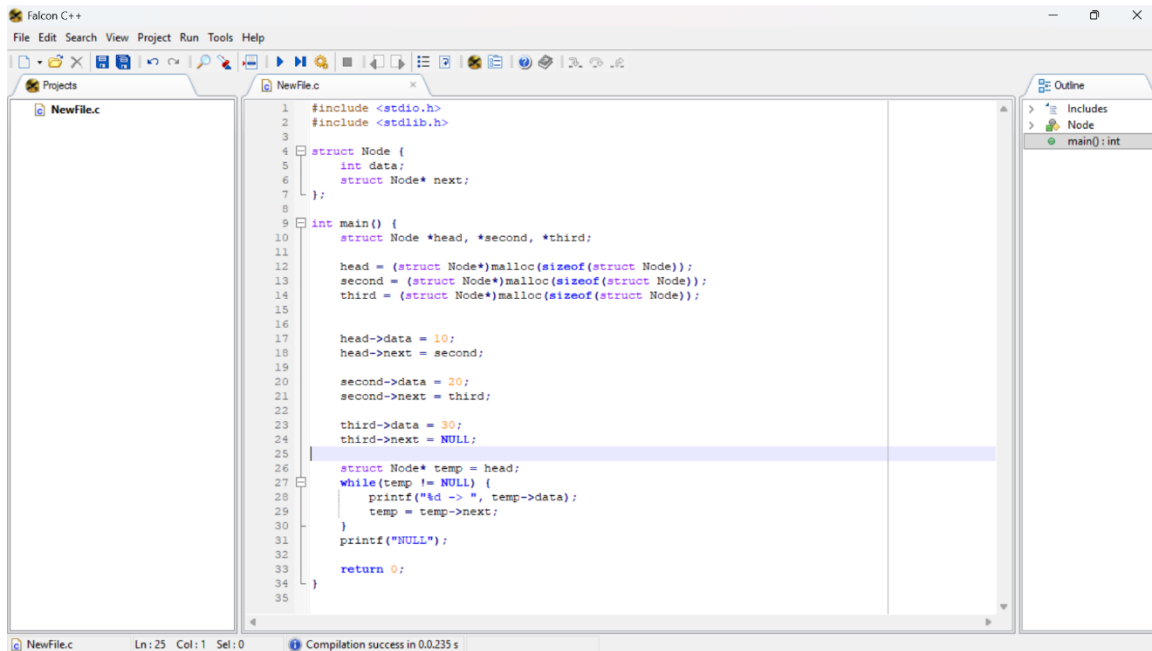
Traversal, insertion, deletion.

### Reasons

Dynamic size, efficient insertion and deletion.

### How They Work

Nodes stored in non-contiguous memory linked by pointers.



### 3. STACK

#### Where Applied

Operating systems, compilers, browsers.

#### Example Applications

Browser navigation, function calls, undo/redo.

#### Algorithms Used

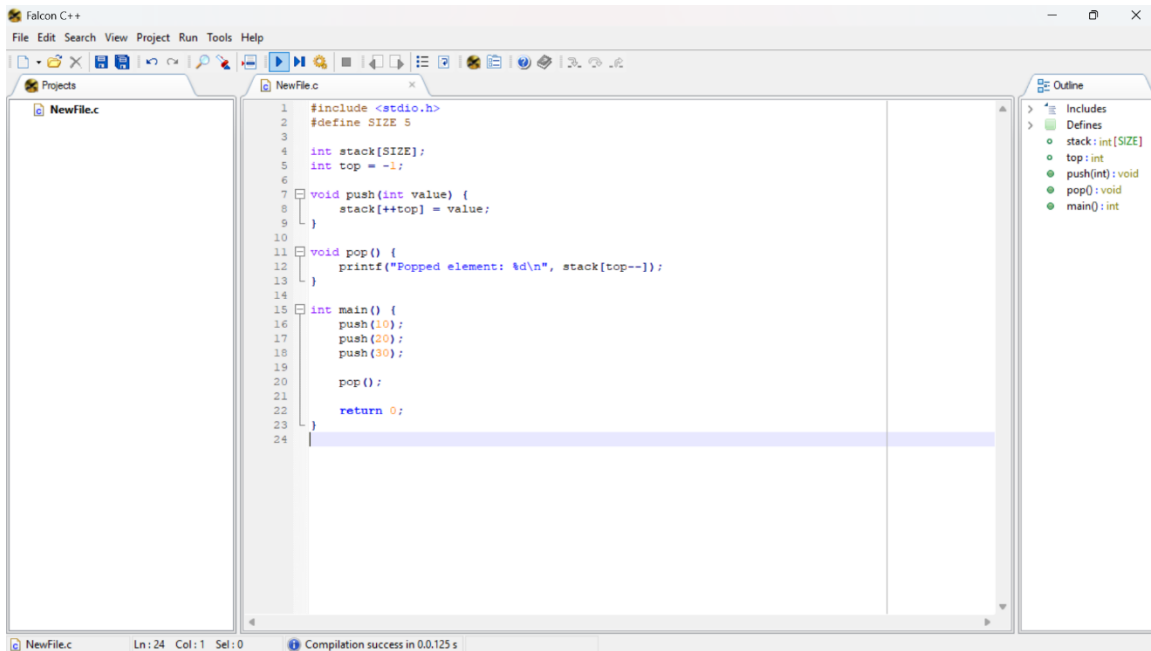
Push, Pop, Peek.

#### Reasons

LIFO principle, fast operations.

#### How They Work

Used in call stacks for function execution.



## 4. QUEUE

### Where Applied

Operating systems, networking, real-time systems.

### Example Applications

Printer spooling, CPU scheduling.

### Algorithms Used

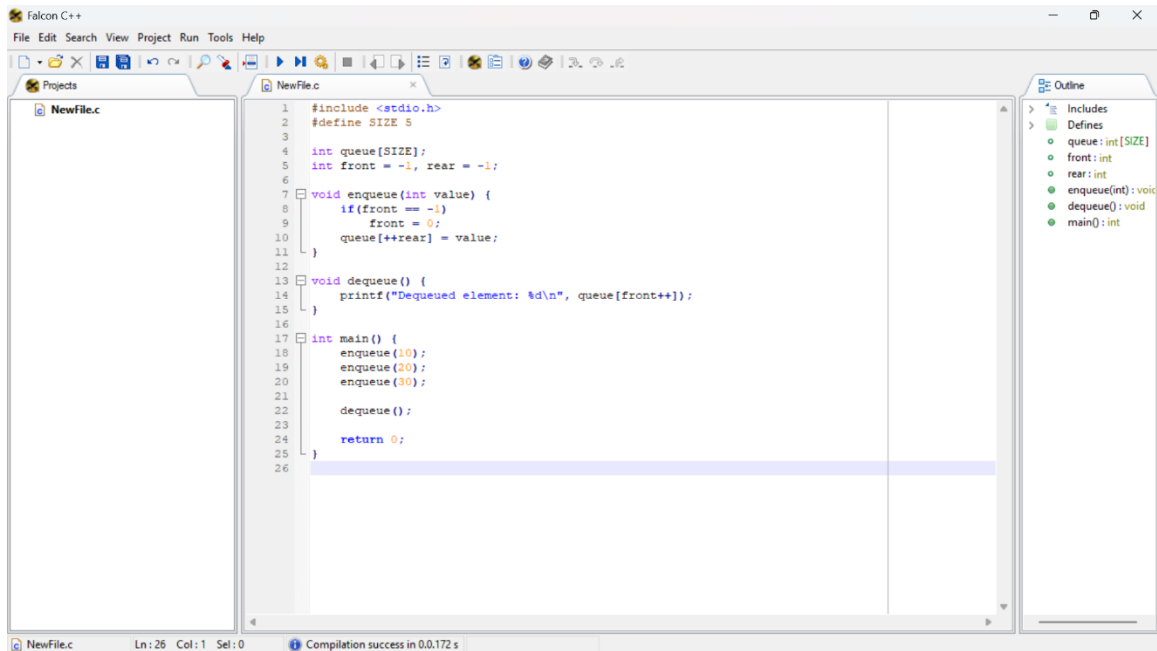
Enqueue, Dequeue.

### Reasons

FIFO principle, fairness.

### How They Work

Manages waiting processes in order.



## 5. TREE

### Where Applied

Databases, file systems, AI.

### Example Applications

Folder structures, database indexing.

### Algorithms Used

Tree traversal, searching.

### Reasons

Hierarchical representation, fast searching.

### How They Work

Balanced trees reduce disk access time.

The screenshot shows a C++ IDE window titled 'Falcon C++'. The main editor displays a C++ program for creating and traversing a binary tree. The code includes headers for `<stdio.h>` and `<stdlib.h>`. It defines a `Node` struct with an `int data` field and pointers to `left` and `right` nodes. A `createNode` function takes an integer value and returns a new node. A `preorder` function performs a preorder traversal, printing the node's data and recursively traversing its left and right children. The `main` function creates a root node with value 1, and its left and right children with values 2 and 3 respectively, then calls `preorder` to traverse the tree.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct Node {
5     int data;
6     struct Node* left;
7     struct Node* right;
8 };
9
10 struct Node* createNode(int value) {
11     struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
12     newNode->data = value;
13     newNode->left = NULL;
14     newNode->right = NULL;
15     return newNode;
16 }
17
18 void preorder(struct Node* root) {
19     if (root != NULL) {
20         printf("%d ", root->data);
21         preorder(root->left);
22         preorder(root->right);
23     }
24 }
25
26 int main() {
27     struct Node* root = createNode(1);
28     root->left = createNode(2);
29     root->right = createNode(3);
30
31     preorder(root);
32     return 0;
33 }
34
```

The right sidebar shows an 'Outline' view with the following structure:

- Includes
- Node
- createNode(int): s
- preorder(struct No
- main(): int

The status bar at the bottom indicates 'Compilation success in 0.0172 s'.

## 6. GRAPH

### Where Applied

Networks, social media, navigation.

### Example Applications

Google Maps, social networks.

### Algorithms Used

BFS, DFS, Dijkstra.

### Reasons

Models relationships, efficient routing.

### How They Work

Represents interconnected systems.

