Name: Hassam Azam Siddiqui Roll no: 02-136221-035
G.M Name: Usman Abubakar Roll no: 02-136221-031

DSA ASSIGNMENT 1

Container ADTs:

A jar where Programmers can bundle related data items into a single unit using the high-level programming concept known as Abstract Data Type (ADT), which makes it simple to manage and manipulate the data. It offers a number of operations and behaviors that can be used to carry out various tasks on the saved data. Since the data is kept in an organized manner that is simple to access and alter, the word "container" is employed.

Containers can be implemented in different ways, and there exist two types of containers; Firstly there are built-in containers like vectors, linked lists and array where as some containers are made up from these built-in containers which are known as ADTs or Abstract Data Types. Stack, queue, priority queue are the examples of ADTs.

Built-in container ADTs are usually implemented using templates and are provided in the standard library class of C++. They can be used to store different type of data in it such as integers, characters, string, objects etc.

Container ADT Operations:

Here we are using vector which is a container ADT and provides wide variety of operations such as :

1. Push_back(): adds a new element in the vector at the end
2. Pop_back(): removes the last element from the vector.
3. Clear(): removes all elements from the vector and makes it clear.
4. Empty(): tells whether the vector is empty or not, return type is Boolean.
5. Size(): returns the size of the vector.
6. Capacity(): returns the number of elements that can be held in the currently allocated storage of memory.

ADT Operations:

In the following program we have a ADT named Studentdata and it contains following functions and members:

Members:

1. Age
2. Roll no
3. First name
4. Last name
5. Grade
6. Section

Name: Hassam Azam Siddiqui Roll no: 02-136221-035
G.M Name: Usman Abubakar Roll no: 02-136221-031

Functions:

1. setAge()
2. setRollNo()
3. setFName()
4. setLName()
5. setGrade()
6. setSection()

Program:

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

// This is ADT
class Studentdata {
private:
	int age,rollNo;
	string fName, lName,grade,section;
public:
	void setAge(int age) {
		this->age = age;
	}
	void setRollNo(int rollNo) {
		this->rollNo = rollNo;
	}
	void setFName(string fName) {
		this->fName = fName;
	}
	void setLName(string lName) {
		this->lName = lName;
	}
	void setGrade(string grade) {
		this->grade = grade;
	}
	void setSection(string section) {
		this->section = section;
	}

	void print() {

		cout << "\t\tName: " << fName << " " << lName << endl;
		cout << "\t\tAge: " << age << endl;
		cout << "\t\tGrade: " << grade << endl;
		cout << "\t\tSection: " << section << endl;
		cout << "\t\tRoll No: " << rollNo << endl;

	}
};
vector <Studentdata> studentdata;//This is vector a container ADT

void addRecord() {
	int noOfEntries;
	int age, rollNo;
	string fName, lName, grade, section;
```

```
	cout << "enter how many entries you wanna perform:";
	cin >> noOfEntries;
	for (int i = 0; i < noOfEntries; i++) {
		Studentdata std;
		cout << "Enter first name of student " << i + 1 << ":";
		cin >> fName;
		std.setFName(fName);
		cout << "Enter last name of student " << i + 1 << ":";
		cin >> lName;
		std.setLName(lName);
		cout << "Enter age of student " << i + 1 << ":";
		cin >> age;
		std.setAge(age);
		cout << "Enter grade of student " << i + 1 << ":";
		cin >> grade;
		std.setGrade(grade);
		cout << "Enter section of student " << i + 1 << ":";
		cin >> section;
		std.setSection(section);
		cout << "Enter roll no of student " << i + 1 << ":";
		cin >> rollNo;
		std.setRollNo(rollNo);

		studentdata.push_back(std);//push_back is a function provided by the
vector this stores the data in the vector
	}
}

void printRecord() {
	if (studentdata.empty()) {// empty() is a function provided by the vector
this returns true or false depending if the vector is empty or have data stored in
it
		cout << "No record found!" << endl;
	}
	else {
		cout << "\t\t\t STUDENT RECORD" << endl;
		cout << "\t\t\t ---------------" << endl;
		cout << endl;
		for (int i = 0; i < studentdata.size(); i++) {// size() is a function
provided by vector this returns the size of vector
			cout << "\t\tStudent " << i + 1 << endl;
			cout << "\t\t--------- " << endl;
			cout << endl;
			studentdata[i].print();
			cout << endl;
		}
	}
}

void checkDatabase() {
	if (studentdata.empty()) {
		cout << "Database is empty!" << endl;
	}
	else
	{
```

```
			cout << "The number of students in database are " <<
studentdata.capacity() << endl;// capacity is a function provided by the vector this
returns how many elements are stored in the memory by vector
			cout << endl;

		}
}

void removeLastRecord() {
		if (studentdata.empty()) {
			cout << "There is no record in database!" << endl;
			cout << endl;
		}
		else {
			studentdata.pop_back();// pop_back is a function provided by the vector
this deletes or remove the latest element in the vector
			cout << "Latest entry removed!" << endl;
			cout << endl;
		}
}

void clearDatabase() {
		if (studentdata.empty()) {
			cout << "There is no record in database!" << endl;
			cout << endl;
		}
		else {
			studentdata.clear();// clear is a function provided by the vector this
clear all the elements stored in a vector
			cout << "All data cleared!" << endl;
			cout << endl;
		}
}

int main() {
		int check;

		bool loop=true;


		while (loop) {

		cout << "\t Sudent Records" << endl;
		cout << "\t ---------------" << endl;
		cout << endl;
		cout << "enter 1 to add student record" << endl;
		cout << "enter 2 to print student record" << endl;
		cout << "enter 3 to check number of students in database" << endl;
		cout << "enter 4 to remove last record" << endl;
		cout << "enter 5 to clear all records" << endl;
		cout << "enter 0 to exit" << endl;

		cin >> check;
		cout << endl;



			switch (check) {
```

```
            case 1:
                  addRecord();
                  break;

            case 2:
                  printRecord();
                  break;

            case 3:
                  checkDatabase();
                  break;

            case 4:
                  removeLastRecord();
                  break;

            case 5:
                  clearDatabase();
                  break;

            case 0:
                  loop = false;
                  break;
            }

      }
      return 0;
}
```

Output:

1

```
enter 2 to print student record
enter 3 to check number of students in database
enter 4 to remove last record
enter 5 to clear all records
enter 0 to exit
1

enter how many entries you wanna perform:2
Enter first name of student 1:Hassam
Enter last name of student 1:Siddiqui
Enter age of student 1:21
Enter grade of student 1:3
Enter section of student 1:A
Enter roll no of student 1:035
Enter first name of student 2:Usman
Enter last name of student 2:Abubakar
Enter age of student 2:20
Enter grade of student 2:3
Enter section of student 2:A
Enter roll no of student 2:031
```

Name: Hassam Azam Siddiqui
G.M Name: Usman Abubakar
Roll no: 02-136221-035
Roll no: 02-136221-031

2

```
         Sudent Records
         --------------

enter 1 to add student record
enter 2 to print student record
enter 3 to check number of students in database
enter 4 to remove last record
enter 5 to clear all records
enter 0 to exit
2

                        STUDENT RECORD
                        --------------

               Student 1
               ---------

               Name: Hassam Siddiqui
               Age: 21
               Grade: 3
               Section: A
               Roll No: 35

               Student 2
               ---------

               Name: Usman Abubakar
               Age: 20
               Grade: 3
               Section: A
               Roll No: 31
```

Name: Hassam Azam Siddiqui
G.M Name: Usman Abubakar
Roll no: 02-136221-035
Roll no: 02-136221-031

3

```
         Sudent Records
         --------------

enter 1 to add student record
enter 2 to print student record
enter 3 to check number of students in database
enter 4 to remove last record
enter 5 to clear all records
enter 0 to exit
3

The number of students in database are 2
```

4

```
         Sudent Records
         --------------

enter 1 to add student record
enter 2 to print student record
enter 3 to check number of students in database
enter 4 to remove last record
enter 5 to clear all records
enter 0 to exit
4

Latest entry removed!
```

Name: Hassam Azam Siddiqui Roll no: 02-136221-035
G.M Name: Usman Abubakar Roll no: 02-136221-031

5

```
         Sudent Records
         --------------

enter 1 to add student record
enter 2 to print student record
enter 3 to check number of students in database
enter 4 to remove last record
enter 5 to clear all records
enter 0 to exit
5

All data cleared!
```