Name : Kunjadiya Brikal Ketanbhai
Roll no : 12
course : MCA
Subject : Object Oriented Concepts and Programming - OOCP

Assignment - 1

1. Write a program to create class Student with student's rollno, name and marks of three subjects (OOCP, AI and MF) and display the details of student with total marks of all subjects along with the percentage in proper format.(Output should be in descending order of percentage.

```
#include <iostream>
using namespace std;

class Student {

    int rollNo;
    string name;
    float oocpMarks, dsMarks, dbmsMarks;
    float totalMarks;
public:
    float percentage;
    Student(int r = 0, string n = "", float oocp = 0, float ds = 0, float dbms = 0) {
        rollNo = r;
        name = n;
        oocpMarks = oocp;
        dsMarks = ds;
        dbmsMarks = dbms;
        totalMarks = oocpMarks + dsMarks + dbmsMarks;
        percentage = (totalMarks / 300) * 100;
    }
    void display() const {
```

```
cout<<rollNo<<"\t"<<name<<"\t"<<oocpMarks<<"\t\t"<<dsMarks<<"\t\t"<<dbmsMarks<<"\t\t"<<to
talMarks<<"\t\t"<<percentage<<"\n";
    }
};
int main() {
    int n;
    cout << "Enter the Number of Students: ";
    cin >> n;

    Student students[n];
    for (int i = 0; i < n; i++) {
        int rollNo;
        string name;
        float oocp, ds, dbms;

        cout << "\nEnter details for Student " << i + 1 << ":\n";
        cout << "Roll No: ";
        cin >> rollNo;
        cout << "Name: ";
        cin.ignore();
        getline(cin, name);
        cout << "Marks in OOCP: ";
        cin >> oocp;
        cout << "Marks in DS: ";
        cin >> ds;
        cout << "Marks in DBMS: ";
        cin >> dbms;
        if(oocp < 0 || oocp > 100 || ds < 0 || ds > 100 || dbms < 0|| dbms > 100){
            printf("Enter Valid Marks");
            exit(0);
```

```
        }
        students[i] = Student(rollNo, name, oocp, ds, dbms);
    }
    int flag=0;
    Student temp;
    for(int i=0;i<n-1;i++){
        for(int j=0;j<n-i-1;j++){
            if(students[j].percentage < students[j+1].percentage){
                flag=1;
            temp = students[j];
            students[j] = students[j+1];
            students[j+1] = temp;
            }
        }
        if(flag==0){
            break;
        }
    }
    cout << "\nStudent details in Descending Order of Percentage:\n";
    cout<<"roll no\t name\t oocp marks\t ds marks\t dbms marks\t total marks\t percentage\n";
    for (int i = 0; i < n; i++) {
        students[i].display();
    }

    return 0;
}
```

Output:-

```
Enter the Number of Students: 3

Enter details for Student 1:
```

Roll No: 1

Name: Mitesh

Marks in OOCP: 60

Marks in DS: 70

Marks in DBMS: 80

Enter details for Student 2:

Roll No: 2

Name: Brikal

Marks in OOCP: 70

Marks in DS: 80

Marks in DBMS: 90

Enter details for Student 3:

Roll No: 3

Name: Vansh

Marks in OOCP: 75

Marks in DS: 85

Marks in DBMS: 95

Student details in Descending Order of Percentage:

roll no	name	oocp marks	ds marks	dbms marks	total marks	percentage
1	Mitesh	60	70	80	210	70.0000
2	Brikal	70	80	90	240	80.0000
3	Vansh	75	85	95	255	85.0000

4. Write a program to create class Date (int day, int month, int year). Read a value as day from user to display new date after adding the value to day in Date.

```
#include <iostream>
using namespace std;
```

```
class Date {

    int Day, Month, Year;

public:
    Date(int d, int m, int y) {
        Day = d;
        Month = m;
        Year = y;
    }
    void addDays(int extraDays) {
        Day += extraDays;
        while (Day > 30) {
            Day -= 30;
            Month++;

            if (Month > 12) {
                Month = 1;
                Year++;
            }
        }
    }

    void display() {
        cout << "Updated Date: " << Day << "/" << Month << "/" << Year << endl;
    }
};

int main() {
    int d, m, y, extraDays;
```

```
    cout << "Enter Day: ";
    cin >> d;
    cout << "Enter Month: ";
    cin >> m;
    cout << "Enter Year: ";
    cin >> y;
    if(d <= 0 || d > 30 || m <= 0 || m > 12 || y <= 0){
        cout<<"Enter valid Date.";
        exit(0);
    }
    Date date(d, m, y);

    cout << "Enter Number of Days to Add: ";
    cin >> extraDays;
    date.addDays(extraDays);
    date.display();

    return 0;
}
```

Output:-

Enter Day: 15

Enter Month: 8

Enter Year: 2021

Enter Number of Days to Add: 965

Updated Date: 20/10/2023

5.Write a program to create class employee with employee's id, name and basic salary. Calculate gross salary for each employee(HRA 20%, DA 30%, OA 10%).

```
#include <iostream>
using namespace std;

class Employee {
    int id;
    string name;
    float basicSalary,HRA,DA,OA,grossSalary;

public:
    Employee(int empId=0, string empName="", float salary=0,float hra=0,float da=0,float oa=0,float gross =0) {
        id = empId;
        name = empName;
        basicSalary = salary;
        HRA = 0.20 * basicSalary;
        DA = 0.30 * basicSalary;
        OA = 0.10 * basicSalary;
        grossSalary = basicSalary + HRA + DA + OA;
    }
    void displayDetails() {
        cout<<"employee id\t name\t Base salary\t HRA\t DA\t OA\t gross salary\n";

cout<<id<<"\t\t"<<name<<"\t"<<basicSalary<<"\t\t"<<HRA<<"\t"<<DA<<"\t"<<OA<<"\t"<<grossSalary<<"\n";
    }
};

int main() {
    int id,n;
```

```
    string name;
    float basicSalary;
    cout <<"Enter the Number of Employee Detail you want to Add: ";
    cin>>n;
    Employee emp[n];
    for(int i=0;i<n;i++){
        cout << "Enter Employee ID: ";
        cin >> id;
        cout << "Enter Employee Name: ";
        cin.ignore();
        getline(cin, name);
        cout << "Enter Basic Salary: ";
        cin >> basicSalary;
        emp[i]= Employee(id, name, basicSalary);
    }
    cout<<"Employee Detail:\n";
    for(int i=0;i<n;i++){
        emp[i].displayDetails();
    }
    return 0;
}
```

Output:-

```
Enter the Number of Employee Detail you want to Add: 2
Enter Employee ID: 1
Enter Employee Name: Mitesh
Enter Basic Salary: 20000
Enter Employee ID: 2
Enter Employee Name: Brikal
Enter Basic Salary: 22000
Employee Detail:
employee id     name    Base salary     HRA     DA      OA      gross salary
```

1 Mitesh 40000 8000 12000 4000 64000

employee id name Base salary HRA DA OA gross salary

2 Brikal 22000 4400 6600 2200 35200

7. Demonstrate the use of static variables in a class by using it to count the number of times the value is being inputted in the program.

```
#include<iostream>
using namespace std;

class InputCounter {
    int Value;
    static int Count;

public:
   void inputValue() {
        cout << "Enter a Value: ";
        cin >> Value;
        Count++;
    }
    static void displayCount() {
        cout << "Number of Values input: " << Count << endl;
    }
};

int InputCounter::Count = 0;

int main() {
    int n;
    cout<<"How many Number you want to Input: ";
```

```
    cin>>n;
    InputCounter obj[n];
    for(int i=0;i<n;i++){
        obj[i].inputValue();
    }
    InputCounter::displayCount();
    return 0;
}
```

Output:-

How many Number you want to Input: 3

Enter a Value: 2

Enter a Value: 4

Enter a Value: 6

Number of Values input: 3

9. Define a class to represent a bank account. Include the following members :

DATA MEMBERS	MEMBER FUNCTIONS
Name of depositor	(1) To assign initial values
Account Number	(2) To Deposit the amount
Type of Account	(3) To withdraw an amount after checking the Balance amount in account
	(4) To display name and balance

Write C++ program to handle 10 customers.

```
#include<iostream>
using namespace std;
class BankAccount{
    int acc_num,balance;
    string name,accountType;
```

```
    public:
    BankAccount(string n = "", int accNum = 0, string accType = "", int initialBalance = 0)
        : name(n), acc_num(accNum), accountType(accType), balance(initialBalance) {}

        void deposit(int amt){
            if(amt >0){
                balance += amt;
            }else{
                cout<<"Enter valid Amount.";
            }
        }
        void withdraw(int amt){
            if(amt > balance){
                cout<<"Balance is too low can not Withdraw";
            }
            else{
                balance = balance - amt;
            }
        }
        void display(){
            cout<<"\nName\t Account Number\t Balance\n";
            cout<<name<<"\t"<<acc_num<<"\t\t"<<balance;
        }
        int getAccountNumber() const {
            return acc_num;
        }
};
int main(){
    int i,choice,n;
    int acc_num,balance;
```

```
    int acc,amt;
    string name,accountType;
    cout<<"How many new Account you want to Create: ";
    cin>>n;
    BankAccount bankaccount[n];
    for(i=0;i<n;i++){
        cout<<"Enter the Account detail for "<<(i+1)<<" User: ";
        cout<<"\nEnter the Name: ";
        cin.ignore();
        getline(cin,name);
        cout<<"Enter the Account Number: ";
        cin>>acc_num;
        cout<<"Enter the Account Type(Saving/Current): ";
        cin.ignore();
        getline(cin,accountType);
        cout<<"Enter the Balance: ";
        cin>>balance;
        bankaccount[i] =  BankAccount(name,acc_num,accountType,balance);
    }
    while(true){
        cout<<"\n1. Deposit the Money.\n2. Withdraw the Money.\n3. Display all data.\n4.
exit\nEnter your choice:";
        cin>>choice;
        bool found = false;
        switch (choice)
        {
        case 1:

            cout<<"Enter the Account Number: ";
            cin>>acc;
            cout<<"Enter the Amount you want to Deposit: ";
```

```
        cin>>amt;
        for(i=0;i<n;i++){
            if(bankaccount[i].getAccountNumber()==acc){
                bankaccount[i].deposit(amt);
                bankaccount[i].display();
                found = true;
                break;
            }
        }
        if(!found){
            cout<<"Bank Account does not Exist.";
        }
        break;
    case 2:
        cout<<"Enter the Account Number: ";
        cin>>acc;
        cout<<"Enter the Amount you want to Withdraw: ";
        cin>>amt;
        for(i=0;i<n;i++){
            if(bankaccount[i].getAccountNumber()==acc){
                bankaccount[i].withdraw(amt);
                bankaccount[i].display();
                found=true;
                break;
                }
            }
            if(!found){
                cout<<"\nBank Account does not Exist.";
            }
        break;
    case 3:
```

```
            for(i=0;i<n;i++){
            bankaccount[i].display();
            }
            break;
        case 4:
            exit(0);
        default:
            cout<<"Enter valid Choice";
            break;
        }
    }
    return 0;
}
```

Output:-

```
How many new Account you want to Create: 2
Enter the Account detail for 1 User:
Enter the Name: Mitesh
Enter the Account Number: 456
Enter the Account Type(Saving/Current): Savings
Enter the Balance: 5000
Enter the Account detail for 2 User:
Enter the Name: Vansh
Enter the Account Number: 123
Enter the Account Type(Saving/Current): Savings
Enter the Balance: 8000

1. Deposit the Money.
2. Withdraw the Money.
3. Display all data.
4. exit
```

Enter your choice:1

Enter the Account Number: 456

Enter the Amount you want to Deposit: 5000

Name Account Number Balance

Mitesh 123 10000

1. Deposit the Money.

2. Withdraw the Money.

3. Display all data.

4. exit

Enter your choice:2

Enter the Account Number: 123

Enter the Amount you want to Withdraw: 2000

Name Account Number Balance

Raj 432 6000

1. Deposit the Money.

2. Withdraw the Money.

3. Display all data.

4. exit

Enter your choice:3

Name Account Number Balance

Mitesh 456 10000

Name Account Number Balance

Vansh 432 6000

1. Deposit the Money.

2. Withdraw the Money.

3. Display all data.

4. exit

Enter your choice:4

Assignment – 2

2) WAP to overload operator * which multiply a number to each element of an array within a class arrayContainer and display the result.

```
#include <iostream>
using namespace std;

class arrayContainer {
private:
    int* arr;
    int size;

public:
    arrayContainer(int s) : size(s) {
        arr = new int[size];
        for (int i = 0; i < size; ++i) {
            arr[i] = 1;
        }
    }

    ~arrayContainer() {
        delete[] arr;
    }

    arrayContainer operator*(int num) {
        arrayContainer result(size);
        for (int i = 0; i < size; ++i) {
            result.arr[i] = this->arr[i] * num;
        }
        return result;
```

```
    }

    void input(){
        cout<<"\nEnter Array Elements : ";
        for(int i = 0; i < size; i++) {
            cin>>arr[i];
        }
    }

    void display() {
        cout<<"\nArray Elements : [ ";
        for (int i = 0; i < size; ++i) {
            cout << arr[i] << " ";
        }
        cout << " ] ";
    }
};

int main() {

    int size,mul;
    cout<<"Enter Size of an Array :- ";
    cin>>size;

    arrayContainer obj(size);
    obj.input();
    cout<<"enter the number to multiply: ";
    cin>>mul;
    if(mul<=0){
        cout<<"multiplyer cannot be zero or less";
        return 0;
```

```
    }else{
        cout <<"\nOriginal array: ";
        obj.display();

        arrayContainer result = obj * 3;
        cout <<"\nArray after multiplication: ";
        result.display();

        return 0;
    }
}
```

Output:-

```
Enter Size of an Array :- 4

Enter Array Elements : 1
2
12
13

enter the number to multiply: 2

Original array:
Array Elements : [  1 2 12 13 ]
Array after multiplication:
Array Elements : [ 2 4 24 26  ]
```

4) WAP to define an object m1 of matrix class, use m1<<cout.

```
#include <iostream>
```

```
using namespace std;

class Matrix
{
    int rows, cols;
    int **data;
public:
    Matrix(int r, int c) : rows(r), cols(c)
    {
        data = new int *[rows];
        for (int i = 0; i < rows; ++i)
        {
            data[i] = new int[cols];
            for (int j = 0; j < cols; ++j)
            {
                data[i][j] = i + j;
            }
        }
    }

    void input()
    {
        cout << "\nEnter Matrix Elements : ";
        for (int i = 0; i < rows; ++i)
        {
            for (int j=0; j < cols; ++j)
            {
                cout << "\nEnter [" << i << "][" << j << "] Element :- ";
                cin >> data[i][j];
            }
        }
```

```
    }

    friend ostream &operator<<(ostream &os, const Matrix &mat);
};

ostream &operator<<(ostream &os, const Matrix &mat)
{
    os << "\nMatrix Elements are :\n";
    for (int i = 0; i < mat.rows; i++)
    {
        for (int j=0; j < mat.cols; j++)
        {
            os<< mat.data[i][j]<<" ";
        }
        os<< endl;
    }
    return os;
}

int main()
{
    int row,col;
    cout<<"\nEnter Row : ";
    cin>>row;
    cout<<"\nEnter Column : ";
    cin>>col;


    Matrix m1(row,col);
    m1.input();
```

```
    cout <<"\nMatrix m1 : " << endl;
    cout << m1;

    return 0;
}
```

Output:-

```
Enter Row : 3
Enter Column : 3
Enter Matrix Elements :
Enter [0][0] Element :- 1
Enter [0][1] Element :- 2
Enter [0][2] Element :- 3
Enter [1][0] Element :- 1
Enter [1][1] Element :- 2
Enter [1][2] Element :- 3
Enter [2][0] Element :- 1
Enter [2][1] Element :- 2
Enter [2][2] Element :- 3
Matrix m1 :
Matrix Elements are :
1 2 3
1 2 3
1 2 3
```

7) WAP to define a class Time with properties int hour; int minute; int second; overload the following operators.

6.1) + operator [a+b] (a is of time type and b is an integer)

6.2) – operator [a-b(same as above)]

6.3) = operator

6.4) <,<=,>,>=

6.5) ++,--[post and pre both]

```
#include <iostream>
using namespace std;

class Time {
private:
    int hour, minute, second;

public:

    Time(int h = 0, int m = 0, int s = 0) : hour(h), minute(m), second(s) {}

    Time operator+(int seconds) const {
        Time newTime = *this;
        newTime.second += seconds;
        while (newTime.second >= 60) {
            newTime.second -= 60;
            newTime.minute++;
            if (newTime.minute >= 60) {
                newTime.minute -= 60;
                newTime.hour++;
                if (newTime.hour >= 24) {
                    newTime.hour -= 24;
                }
            }
        }
        return newTime;
```

```
}

Time operator-(int seconds) const {
    Time newTime = *this;
    newTime.second -= seconds;
    while (newTime.second < 0) {
        newTime.second += 60;
        newTime.minute--;
        if (newTime.minute < 0) {
            newTime.minute += 60;
            newTime.hour--;
            if (newTime.hour < 0) {
                newTime.hour += 24;
            }
        }
    }
    return newTime;
}

Time& operator=(const Time& other) {
    if (this != &other) {
        hour = other.hour;
        minute = other.minute;
        second = other.second;
    }
    return *this;
}

bool operator<(const Time& other) const {
    if (hour < other.hour) return true;
    if (hour == other.hour && minute < other.minute) return true;
```

```
        if (hour == other.hour && minute == other.minute && second < other.second) return
true;
        return false;
    }

    bool operator<=(const Time& other) const {
        return *this < other || *this == other;
    }

    bool operator>(const Time& other) const {
        return !(*this <= other);
    }

    bool operator>=(const Time& other) const {
        return !(*this < other);
    }


    bool operator==(const Time& other) const {
        return (hour == other.hour && minute == other.minute && second == other.second);
    }

    bool operator!=(const Time& other) const {
        return !(*this == other);
    }

    Time& operator++() {
        *this = *this + 1;
        return *this;
    }
```

```
    Time operator++(int) {
        Time temp = *this;
        ++(*this);
        return temp;
    }

    Time& operator--() {
        *this = *this - 1;
        return *this;
    }

    Time operator--(int) {
        Time temp = *this;
        --(*this);
        return temp;
    }

    void display() const {
        cout << (hour < 10 ? "0" : "") << hour << ":"
             << (minute < 10 ? "0" : "") << minute << ":"
             << (second < 10 ? "0" : "") << second << endl;
    }
};

int main() {
    Time t1(10, 30, 45);    Time t2;

    t2 = t1 + 75;
    cout << "t1 + 75 seconds: ";
    t2.display();
```

```
    t2 = t1 - 3605;
    cout << "t1 - 70 seconds: ";
    t2.display();

    cout << "t1 == t2: " << (t1 == t2 ? "True" : "False") << endl;
    cout << "t1 != t2: " << (t1 != t2 ? "True" : "False") << endl;
    cout << "t1 < t2: " << (t1 < t2 ? "True" : "False") << endl;
    cout << "t1 <= t2: " << (t1 <= t2 ? "True" : "False") << endl;
    cout << "t1 > t2: " << (t1 > t2 ? "True" : "False") << endl;
    cout << "t1 >= t2: " << (t1 >= t2 ? "True" : "False") << endl;

    cout << "Pre-increment t1 : ";
    (++t1).display();

    cout << "Post-increment t1 : ";
    (t1++).display();
    t1.display();

    cout << "Pre-decrement t1 : ";
    (--t1).display();

    cout << "Post-decrement t1 : ";
    (t1--).display();
    t1.display();

    return 0;
}
```

Output:-

```
t1 + 75 seconds: 10:32:00
```

t1 - 3605 seconds: 09:30:40

t1 == t2: False

t1 != t2: True

t1 < t2: False

t1 <= t2: False

t1 > t2: True

t1 >= t2: True

Pre-increment t1 : 10:30:46

Post-increment t1 : 10:30:46

10:30:47

Pre-decrement t1 : 10:30:46

Post-decrement t1 : 10:30:46

10:30:45

14) Write a generic function that will sort a character string, integer and float value. Create a menu with appropriate options and accept the values from the user.

```
#include <iostream>
using namespace std;

template <typename T>
void sortArray(T arr[], int n)
{
   for (int i = 0; i < n - 1; ++i)
   {
      for (int j = 0; j < n - 1 - i; ++j)
      {
         if (arr[j] > arr[j + 1])
         {
```

```
                T temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

template <typename T>
void printArray(T arr[], int n)
{
    for (int i = 0; i < n; ++i)
    {
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main()
{
    int choice;
    do
    {
        cout << "\n---------- MAIN Menu ----------:\n";
        cout << "1. Sort Integer Array\n";
        cout << "2. Sort Float Array\n";
        cout << "3. Sort Character String\n";
        cout << "4. Exit\n";
        cout << "\nEnter your choice: ";
        cin >> choice;
```

```
switch (choice)
{
case 1:
{
    int n;
    cout << "\nEnter the number of elements: ";
    cin >> n;
    int arr[n];

    cout << "\nEnter the integer elements:\n";
    for (int i = 0; i < n; ++i)
    {
        cin >> arr[i];
    }

    sortArray(arr, n);
    cout << "\nSorted Integer Array: ";
    printArray(arr, n);
    break;
}
case 2:
{
    int n;
    cout << "\nEnter the number of elements: ";
    cin >> n;
    float arr[n];

    cout << "\nEnter the float elements:\n";
    for (int i = 0; i < n; ++i)
    {
        cin >> arr[i];
```

```
            }
            sortArray(arr, n);
            cout << "\nSorted Float Array: ";
            printArray(arr, n);
            break;
        }
        case 3:
        {
            string str;
            cout << "\nEnter a string: ";
            cin >> str;
            int n = str.length();
            char arr[n + 1];

            for (int i = 0; i < n; ++i)
            {
                arr[i] = str[i];
            }

            sortArray(arr, n);
            cout << "\nSorted String: ";
            printArray(arr, n);
            break;
        }
        case 4:
            cout << "\nExiting the program." << endl;
            break;
        default:
            cout << "\nInvalid choice! Please try again." << endl;
        }
    } while (choice != 4);
```

```
    return 0;
}
```

Output:-

```
|| MAIN Menu ||:
1. Sort Integer Array
2. Sort Float Array
3. Sort Character String
4. Exit

Enter your choice: 1
Enter the number of elements: 4
Enter the integer elements:
2
10
7
18
Sorted Integer Array: 2 7 10 18

|| MAIN Menu ||:
1. Sort Integer Array
2. Sort Float Array
3. Sort Character String
4. Exit

Enter your choice: 2
Enter the number of elements: 3
Enter the float elements:
2.456
4.10
```

1.12

Sorted Float Array: 1.12 2.456 4.10

|| MAIN Menu ||:

1. Sort Integer Array

2. Sort Float Array

3. Sort Character String

4. Exit

Enter your choice: 3

Enter a string: Vansh

Sorted String: a h n s v

|| MAIN Menu ||:

1. Sort Integer Array

2. Sort Float Array

3. Sort Character String

4. Exit

Enter your choice: 4

Exiting the program.

15) Write a template function called find(). This function searches an array for an object. It returns either the index of the matching object (if one is found) or -1 if no match is found.

```
#include <iostream>
using namespace std;

template <typename T>
```

```
int find(const T arr[], int size, const T& target) {
    for (int i = 0; i < size; ++i) {
        if (arr[i] == target) {
            return i; // Return index if found
        }
    }
    return -1; // Return -1 if not found
}

void findInIntegerArray() {
    const int size = 5;
    int arr[size];

    cout << "Enter 5 integers:\n";
    for (int i = 0; i < size; ++i) {
        cin >> arr[i];
    }

    int target;
    cout << "Enter the integer to search for: ";
    cin >> target;

    int result = find(arr, size, target);
    if (result != -1)
        cout << "Found at index " << result << endl;
    else
        cout << "Not found" << endl;
}

void findInFloatArray() {
    const int size = 5;
```

```
    float arr[size];

    cout << "Enter 5 float values:\n";
    for (int i = 0; i < size; ++i) {
        cin >> arr[i];
    }

    float target;
    cout << "Enter the float value to search for: ";
    cin >> target;

    int result = find(arr, size, target);
    if (result != -1)
        cout << "Value found at index " << result << endl;
    else
        cout << "Value not found" << endl;
}

void findInString() {
    const int size = 5;
    string input;

    cout << "Enter a string of exactly 5 characters: ";
    cin >> input;

    if (input.length() != size) {
        cout << "Error: String must be exactly 5 characters long.\n";
        return;
    }

    char target;
```

```
    cout << "Enter the character to search for: ";
    cin >> target;

    int result = find(input.c_str(), size, target); // Use input as a C-string
    if (result != -1)
        cout << "Character found at index " << result << endl;
    else
        cout << "Character not found" << endl;
}

int main() {
    int choice;

    do {
        cout << "\nMenu:\n";
        cout << "1. Search in integer array\n";
        cout << "2. Search in float array\n";
        cout << "3. Search in string array\n";
        cout << "4. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                findInIntegerArray();
                break;
            case 2:
                findInFloatArray();
                break;
            case 3:
                findInString();
```

```
            break;
        case 4:
            cout << "Exiting program. \n";
            break;
        default:
            cout << "Invalid choice! Please try again.\n";
      }
   } while (choice != 4);

   return 0;
}
```

Output:-

```
Menu:
1. Search in integer array
2. Search in float array
3. Search in string array
4. Exit
Enter your choice: 2
Enter 5 float values:
10.12
1.10
5.20
3.40
2.10
Enter the float value to search for: 1.12
Value not found

Menu:
1. Search in integer array
```

2. Search in float array

3. Search in string array

4. Exit

Enter your choice: 3

Enter a string of exactly 5 characters: Mitesh

Enter the character to search for: t

Character found at index 2

Menu:

1. Search in integer array

2. Search in float array

3. Search in string array

4. Exit

Enter your choice: 1

Enter 5 integers:

12

9

7

1

8

Enter the integer to search for: 4

Not found

Menu:

1. Search in integer array

2. Search in float array

3. Search in string array

4. Exit

Enter your choice: 4

Exiting program.