Labsheet-08

Submitted by- Vineet Giri
Submitted to- Mr Naman Chauhan
BCA year: 1st
Section: C

Program List

1. Write a C++ function to find a prime number.

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

bool isPrime(int n) {
    if (n <= 1) return false;
    for (int i = 2; i <= n / 2; i++) {
        if (n % i == 0)
            return false;
```

```
    }
    return true;
}

int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;
    if (isPrime(num))
        cout << num << " is a Prime Number.";
    else
        cout << num << " is not a Prime Number.";
    return 0;
}
```

Output:

```
Enter a number: 7
7 is a Prime Number.
```

2. Write a c++ function to print the Fibonacci series.

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

void fibonacci(int n) {
    int a = 0, b = 1, c;
```

```
    cout << "Fibonacci Series: ";
    for (int i = 0; i < n; i++) {
        cout << a << " ";
        c = a + b;
        a = b;
        b = c;
    }
}

int main() {
    int n;
    cout << "Enter number of terms: ";
    cin >> n;
    fibonacci(n);
    return 0;
}
```

Output:

```
Enter number of terms: 6
Fibonacci Series: 0 1 1 2 3 5
```

3. Write a C++ function to apply the bubble sort technique on an array.

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

```
void bubbleSort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1])
                swap(arr[j], arr[j + 1]);
        }
    }
}

int main() {
    int n;
    cout << "Enter array size: ";
    cin >> n;
    int arr[n];
    cout << "Enter elements: ";
    for (int i = 0; i < n; i++) cin >> arr[i];
    bubbleSort(arr, n);
    cout << "Sorted array: ";
    for (int i = 0; i < n; i++) cout << arr[i] << " ";
    return 0;
}
```

Output:

```
Enter array size: 5
Enter elements: 5 1 4 2 3
Sorted array: 1 2 3 4 5
```

4. Write a C++ function to copy a string from another string.

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

void copyString(char s1[], char s2[]) {
    strcpy(s2, s1);
}

int main() {
    char str1[50], str2[50];
    cout << "Enter a string: ";
    cin.getline(str1, 50);
    copyString(str1, str2);
    cout << "Copied string: " << str2;
    return 0;
}
```

Output:

```
Enter a string: Vineet Giri
Copied string: Vineet Giri
```

5. Write a C++ function to find out the number of words in a string.

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

int countWords(string str) {
    int count = 0;
    bool inWord = false;
    for (char ch : str) {
        if (isspace(ch))
            inWord = false;
        else if (!inWord) {
            inWord = true;
            count++;
        }
    }
    return count;
}

int main() {
    string str;
    cout << "Enter a sentence: ";
    getline(cin, str);
    cout << "Number of words: " << countWords(str);
    return 0;
}
```

Output:

```
Enter a sentence: Hello my name is Vineet
```

Number of words: 5

6. Write a C++ function to search and element using binary search.

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

int binarySearch(int arr[], int n, int key) {
    int low = 0, high = n - 1;
    while (low <= high) {
        int mid = (low + high) / 2;
        if (arr[mid] == key) return mid;
        else if (arr[mid] < key) low = mid + 1;
        else high = mid - 1;
    }
    return -1;
}

int main() {
    int n, key;
    cout << "Enter size of array: ";
    cin >> n;
    int arr[n];
    cout << "Enter sorted elements: ";
    for (int i = 0; i < n; i++) cin >> arr[i];
    cout << "Enter element to search: ";
    cin >> key;
```

```
    int result = binarySearch(arr, n, key);
    if (result != -1)
        cout << "Element found at index " << result;
    else
        cout << "Element not found.";
    return 0;
}
```

Output:

```
Enter size of array: 5
Enter sorted elements: 10 20 30 40 50
Enter element to search: 30
Element found at index 2
```

7. Write a c++ function to find the sum of two Matrices.

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

int main() {
    int a[2][2], b[2][2], sum[2][2];
    cout << "Enter elements of first matrix:\n";
    for (int i = 0; i < 2; i++)
        for (int j = 0; j < 2; j++)
            cin >> a[i][j];
```

```
    cout << "Enter elements of second matrix:\n";
    for (int i = 0; i < 2; i++)
        for (int j = 0; j < 2; j++)
            cin >> b[i][j];

    cout << "Sum of matrices:\n";
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            sum[i][j] = a[i][j] + b[i][j];
            cout << sum[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

Output:

```
Enter elements of first matrix:
1 2
3 4
Enter elements of second matrix:
5 6
7 8
Sum of matrices:
6 8
10 12
```

8. Write a C++ function to find out the sum of diagonal numbers in a matrix.

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

int main() {
    int a[3][3], sum = 0;
    cout << "Enter elements of matrix:\n";
    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            cin >> a[i][j];

    for (int i = 0; i < 3; i++)
        sum += a[i][i];

    cout << "Sum of diagonal elements: " << sum;
    return 0;
}
```

Output:

```
Enter elements of matrix:
1 2 3
4 5 6
7 8 9
Sum of diagonal elements: 15
```

9. Write a C++ function to swap two numbers using call-by-reference.

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

void swapNumbers(int &x, int &y) {
    int temp = x;
    x = y;
    y = temp;
}

int main() {
    int a, b;
    cout << "Enter two numbers: ";
    cin >> a >> b;
    swapNumbers(a, b);
    cout << "After swapping: a = " << a << ", b = " <<
b;
    return 0;
}
```

Output:

```
Enter two numbers: 10 20
After swapping: a = 20, b = 10
```

10. Write a program to define a class Student with data members name, roll_no, and marks.

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

class Student {
    string name;
    int roll_no;
    float marks;

public:
    void input() {
        cout << "Enter name: ";
        cin >> name;
        cout << "Enter roll number: ";
        cin >> roll_no;
        cout << "Enter marks: ";
        cin >> marks;
    }
    void display() {
        cout << "\nName: " << name << "\nRoll No: "
<< roll_no << "\nMarks: " << marks;
    }
};

int main() {
```

```
    Student s;
    s.input();
    s.display();
    return 0;
}
```

Output:

```
Enter name: Vineet
Enter roll number: 101
Enter marks: 88
Name: Vineet
Roll No: 101
Marks: 88
```

11. Write a program to define a class Car member's brand and price.

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

class Car {
public:
    string brand;
    float price;

    void input() {
        cout << "Enter car brand: ";
```

```
        cin >> brand;
        cout << "Enter car price: ";
        cin >> price;
    }

    void display() {
        cout << "Brand: " << brand << " | Price: " <<
price << endl;
    }
};

int main() {
    Car c1, c2;
    cout << "Enter details for first car:\n";
    c1.input();
    cout << "Enter details for second car:\n";
    c2.input();

    cout << "\n--- Car Details ---\n";
    c1.display();
    c2.display();
    return 0;
}
```

Output:

```
Enter details for first car:
Enter car brand: BMW
Enter car price: 5500000
```

```
Enter details for second car:
Enter car brand: Audi
Enter car price: 6000000

--- Car Details ---
Brand: BMW | Price: 5500000
Brand: Audi | Price: 6000000
```

12. Write a program to demonstrate encapsulation using a class Bank account.

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

class BankAccount {
private:
    float balance;

public:
    BankAccount() { balance = 0; }

    void deposit(float amount) {
        balance += amount;
        cout << "Deposited: " << amount << endl;
    }

    void withdraw(float amount) {
```

```
        if (amount <= balance) {
            balance -= amount;
            cout << "Withdrawn: " << amount << endl;
        } else
            cout << "Insufficient balance!" << endl;
    }

    void showBalance() {
        cout << "Current Balance: " << balance <<
endl;
    }
};

int main() {
    BankAccount b;
    b.deposit(5000);
    b.withdraw(2000);
    b.showBalance();
    return 0;
}
```

Output:

```
Deposited: 5000
Withdrawn: 2000
Current Balance: 3000
```

13. Write a program to demonstrate single inheritance.

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

class Person {
public:
    string name;
    int age;
    void getData() {
        cout << "Enter name: ";
        cin >> name;
        cout << "Enter age: ";
        cin >> age;
    }
};

class Student : public Person {
public:
    float marks;
    void getMarks() {
        cout << "Enter marks: ";
        cin >> marks;
    }
    void display() {
        cout << "\nName: " << name << "\nAge: " <<
age << "\nMarks: " << marks << endl;
```

```
    }
};

int main() {
    Student s;
    s.getData();
    s.getMarks();
    s.display();
    return 0;
}
```

Output:

```
Enter name: Vineet
Enter age: 18
Enter marks: 92

Name: Vineet
Age: 18
Marks: 92
```

14. Write a program to demonstrate function overloading (compile-time polymorphism).

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

```
class Calculator {
public:
    int add(int a, int b) {
        return a + b;
    }
    double add(double a, double b) {
        return a + b;
    }
};

int main() {
    Calculator c;
    cout << "Sum (int): " << c.add(5, 10) << endl;
    cout << "Sum (double): " << c.add(2.5, 3.7) <<
endl;
    return 0;
}
```

Output:

```
Sum (int): 15
Sum (double): 6.2
```

15. Write a program to demonstrate abstraction using an abstract class Shape.

```
#include <iostream>
```

```
using namespace std;

class Shape {
public:
    virtual void area() = 0; // Pure virtual function
};

class Circle : public Shape {
    float radius;
public:
    Circle(float r) { radius = r; }
    void area() {
        cout << "Area of Circle: " << 3.14 * radius *
radius << endl;
    }
};

class Rectangle : public Shape {
    float length, breadth;
public:
    Rectangle(float l, float b) {
        length = l;
        breadth = b;
    }
    void area() {
        cout << "Area of Rectangle: " << length *
breadth << endl;
    }
};
```

```
int main() {
    Circle c(5);
    Rectangle r(4, 6);
    c.area();
    r.area();
    return 0;
}
```

Output:

```
Area of Circle: 78.5
Area of Rectangle: 24
```