ASSESSMENT 2

1. Write a C++ program to find the most frequent element in an array

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

int main() {
    vector<int> arr = {1, 3, 2, 3, 4, 1, 3, 2, 1, 1};
    unordered_map<int, int> freq;
    int mostFrequent = arr[0];
    int maxCount = 1;

    for (int num : arr) {
        freq[num]++;
        if (freq[num] > maxCount) {
            maxCount = freq[num];
            mostFrequent = num;
        }
    }

     cout << "The most frequent element is: " << mostFrequent << " (Frequency: " << maxCount
<< ")" << endl;
    return 0;
}
```

Output :

The most frequent element is: 1 (Frequency: 4)

2. Create a class for addition of three numbers using operator overloading

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

class Numbers {
private:
    int a, b, c;
public:
    Numbers(int x = 0, int y = 0, int z = 0) : a(x), b(y), c(z) {}
```

```
    Numbers operator+(const Numbers &n) {
        return Numbers(a + n.a, b + n.b, c + n.c);
    }

    void display() {
        cout << "Sum: " << a << ", " << b << ", " << c << endl;
    }
};

int main() {
    Numbers num1(1, 2, 3);
    Numbers num2(4, 5, 6);
    Numbers sum = num1 + num2;

    sum.display();
    return 0;
}
```

Output :
Sum: 5, 7, 9

3. Generate C++ program to demonstrate multiple inheritance by creating a class cuboid which extends class rectangle and Shape to calculate its area and perimeter

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

class Shape {
public:
    virtual void area() = 0;
    virtual void perimeter() = 0;
};

class Rectangle : public Shape {
protected:
    int length, breadth;
public:
    Rectangle(int l, int b) : length(l), breadth(b) {}

    void area() override {
```

```
        cout << "Area of Rectangle: " << length * breadth << endl;
    }

    void perimeter() override {
        cout << "Perimeter of Rectangle: " << 2 * (length + breadth) << endl;
    }
};

class Cuboid : public Rectangle {
private:
    int height;
public:
    Cuboid(int l, int b, int h) : Rectangle(l, b), height(h) {}

    void area() override {
         cout << "Surface Area of Cuboid: " << 2 * (length * breadth + breadth * height + height *
length) << endl;
    }

    void perimeter() override {
        cout << "Perimeter of Cuboid (sum of edges): " << 4 * (length + breadth + height) << endl;
    }

    void volume() {
        cout << "Volume of Cuboid: " << length * breadth * height << endl;
    }
};

int main() {
    Cuboid cub(2, 3, 4);
    cub.area();
    cub.perimeter();
    cub.volume();
    return 0;
}
```

Output :

Surface Area of Cuboid: 52
Perimeter of Cuboid (sum of edges): 36
Volume of Cuboid: 24

4. Create a class Number and the derived Skipper, write a program to print the m to n numbers by skipping k numbers in between using appropriate function and variable

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

class Number {
protected:
    int m, n, k;
public:
    Number(int start, int end, int skip) : m(start), n(end), k(skip) {}

    virtual void printNumbers() {
        for (int i = m; i <= n; i++) {
            cout << i << " ";
        }
        cout << endl;
    }
};

class Skipper : public Number {
public:
    Skipper(int start, int end, int skip) : Number(start, end, skip) {}

    void printNumbers() override {
        for (int i = m; i <= n; i += (k + 1)) {
            cout << i << " ";
        }
        cout << endl;
    }
};

int main() {
    int m = 1, n = 10, k = 2;
    Skipper skip(m, n, k);

    cout << "Numbers from " << m << " to " << n << " skipping " << k << " numbers:" << endl;
    skip.printNumbers();
```

```
    return 0;
}
```

Output :

```
Numbers from 1 to 10 skipping 2 numbers:
1 4 7 10
```