// 01 - This example shows how read string and integer from terminal input.

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

int main(int argc, char const *argv[]) {
    string name = argv[1];
    int year = atoi(argv[2]);

    int age = 2019 - year;
    cout << "Hi " << name << " " << age << " years old." << endl;
    return 0;
}
```

Compile and run with
- g++ a.cpp -o output
- ./output Alex 2000

// 02 - Vector - Size of arrays are fixed whereas the vectors are resizable.

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

int main(int argc, char const *argv[]) {
    vector<int> v;
    for (int i=1; i<=20; i++)
        v.push_back(i);

    for(auto j : v)
        cout << j << endl;

    cout << "Vector Size = " << v.size() << '\n';
    return 0;
}
```

// 03 - Using Structure

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

struct Person {
    char name[50];
    int age;
    float salary;
};

int main() {
    Person p1;
```
cout << "Enter Full name: ";
cin.get(p1.name, 50);
cout << "Enter age: ";
cin >> p1.age;
cout << "Enter salary: ";
cin >> p1.salary;
cout <<  
Displaying Information. " << endl;
cout << "Name: " << p1.name << endl;
cout << "Age: " << p1.age << endl;
cout << "Salary: " << p1.salary << endl;

return 0;
}  

//04 - Using Class
#include <iostream>
#include <cstring>
using namespace std;
class Car {
 private:  
 int speed = 30;

 public:  
 string name;

 void speedUp(int n) {
 speed += n;
 cout << "Speed up to " << speed << endl;
 }

 int getSpeed() {
 return speed;
 }
};

int main() {
 Car a;
 a.name = "Toyota";
cout << "Current Speed: " << a.getSpeed() << endl;
a.speedUp(10);
cout << "Current Speed: " << a.getSpeed() << endl;
return 0;
}

//05 - Max item in a Array
#include <iostream>
using namespace std;

int main() {
 int i, n;
 float arr[100];
cout << "Enter total number of elements(1 to 100): ";
cin >> n;
cout << endl;
// Store number entered by the user
for (i = 0; i < n; ++i) {
cout << "Enter Number " << i + 1 << " : ";
cin >> arr[i];
}
// Loop to store largest number to arr[0]
for (i = 1; i < n; ++i) {  
    // Change < to > if you want to find the smallest element 
    if (arr[0] < arr[i]) 
        arr[0] = arr[i];
}  
cout << "Largest element = " << arr[0] << endl; 
return 0;
}
===================================================================================
//06 - Display address of array elements using pointer notation
#include <iostream>
using namespace std;
int main() {
    float arr[5];

    cout << "Displaying address using pointers notation: " << endl;
    for (int i = 0; i < 5; ++i) {  
        cout << arr + i << endl;
    }
    return 0;
}
===================================================================================
//07 - Passing by reference using pointers
#include <iostream>
using namespace std;
// Function prototype 
void swap(int *, int *);
int main() {
    int a = 1, b = 2;
    cout << "Before swapping" << endl;
    cout << "a = " << a << endl;
    cout << "b = " << b << endl;
    swap( & a, & b);
    cout << "\nAfter swapping" << endl;
    cout << "a = " << a << endl;
    cout << "b = " << b << endl;
    return 0;
}
void swap(int * n1, int * n2) {
int temp;
    temp = * n1;
    * n1 = * n2;
    * n2 = temp;
}
===================================================================================
//08 - Enumeration Type
#include <iostream>
using namespace std;
enum week {
    Sunday,
    Monday,
    Tuesday,
    Wednesday,
    Thursday,
    Friday,
    Saturday
};
int main() {
    week today;
    today = Wednesday;
    cout << "Day " << today + 1 << endl;
    return 0;
}

#include <iostream>

using namespace std;
int main() {
    int var1 = 3;
    int var2 = 24;
    int var3 = 17;
    cout << & var1 << endl;
    cout << & var2 << endl;
    cout << & var3 << endl;
}

//09 - Pointer: Address in C++