



Lesson #02 - Code

```
#include<iostream>

using namespace std;

int MySum(int a, int b)
{
    int s = 0;
    s = a + b;
    return s;
}

int main()
{
    int arr1[5] = { 200,100,50,25,30 };
    int a, b, c;

    a = 10;
    b = 20;
    a++;
    ++b;

    c = a + b;

    cout << a << endl;
    cout << b << endl;
    cout << c << endl;

    for (int i = 1; i <= 5; i++)
    {
        cout << i << endl;
        a = a + a * i;
    }

    c = MySum(a, b);
    cout << c;
    return 0;
}
```



Lesson #03 - Code

```
#include<iostream>

using namespace std;

int MySum(int a, int b)
{
    int s = 0;
    s = a + b;
    return s;
}

int main()
{
    int arr1[5] = { 200,100,50,25,30 };
    int a, b, c;

    a = 10;
    b = 20;
    a++;
    ++b;

    c = a + b;

    cout << a << endl;
    cout << b << endl;
    cout << c << endl;

    for (int i = 1; i <= 5; i++)
    {
        cout << i << endl;
        a = a + a * i;
    }

    c = MySum(a, b);
    cout << c;
    return 0;
}
```



Lesson #04 - Code

```
#include<iostream>

using namespace std;

int MySum(int a, int b)
{
    int s = 0;
    s = a + b;
    return s;
}

int main()
{
    int arr1[5] = { 200,100,50,25,30 };
    int a, b, c;

    a = 10;
    b = 20;
    a++;
    ++b;

    c = a + b;

    cout << a << endl;
    cout << b << endl;
    cout << c << endl;

    for (int i = 1; i <= 5; i++)
    {
        cout << i << endl;
        a = a + a * i;
    }

    c = MySum(a, b);
    cout << c;
    return 0;
}
```



Lesson #05 - Code

```
#include<iostream>

using namespace std;

int MySum(int a, int b)
{
    int s = 0;
    s = a + b;
    return s;
}

int main()
{
    int arr1[5] = { 200,100,50,25,30 };
    int a, b, c;

    a = 10;
    b = 20;
    a++;
    ++b;

    c = a + b;

    cout << a << endl;
    cout << b << endl;
    cout << c << endl;

    for (int i = 1; i <= 5; i++)
    {
        cout << i << endl;
        a = a + a * i;
    }

    c = MySum(a, b);
    cout << c;
    return 0;
}
```



Lesson #06 - Code

```
#include<iostream>

using namespace std;

int MySum(int a, int b)
{
    int s = 0;
    s = a + b;
    return s;
}

int main()
{
    int arr1[5] = { 200,100,50,25,30 };
    int a, b, c;

    a = 10;
    b = 20;
    a++;
    ++b;

    c = a + b;

    cout << a << endl;
    cout << b << endl;
    cout << c << endl;

    for (int i = 1; i <= 5; i++)
    {
        cout << i << endl;
        a = a + a * i;
    }

    c = MySum(a, b);
    cout << c;
    return 0;
}
```



Lesson #07 - Code

```
#include<iostream>

using namespace std;

int MySum(int a, int b)
{
    int s = 0;
    s = a + b;
    return s;
}

int main()
{
    int arr1[5] = { 200,100,50,25,30 };
    int a, b, c;

    a = 10;
    b = 20;
    a++;
    ++b;

    c = a + b;

    cout << a << endl;
    cout << b << endl;
    cout << c << endl;

    for (int i = 1; i <= 5; i++)
    {
        cout << i << endl;
        a = a + a * i;
    }

    c = MySum(a, b);
    cout << c;
    return 0;
}
```



Lesson #09 Homework 1 - Solution

```
#include<iostream>

using namespace std;

int main()
{
    int Number = 1;
    string Result;

    Result = (Number > 0) ? "Positive" : "Negative";

    cout << "Number is " << Result << endl;

    return 0;
}
```



Lesson #09 Homework 2 - Solution

```
#include<iostream>

using namespace std;

int main()
{
    int Number = 0;
    string Result;

    Result = (Number == 0) ? "Zero" : ((Number > 0) ? "Positive" :
    "Negative");

    cout << "Number is " << Result;

    return 0;
}
```



Lesson #09 Code

```
#include<iostream>

using namespace std;

int main()
{
    int Mark = 90;
    string result;
    //Using normal if

    if (Mark >= 50)
    {
        result = "PASS";
    }
    else
    {
        result = "FAIL";
    }

    cout << result << endl;

    //Using Short Hand IF

    result = (Mark >= 50) ? "PASS" : "FAIL";

    cout << result << endl;

    return 0;
}
```



Lesson #10 Code

```
#include<iostream>

using namespace std;

int main()
{
    int Array1[] = { 1, 2, 3, 4 };

    for (int n : Array1) {
        cout << n << endl;
    }

    return 0;
}
```



Lesson #11 Code

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

int ReadNumber()
{
    int Number;
    cout << "Please enter a number?" << endl;
    cin >> Number;

    while (cin.fail())
    {
        // user didn't input a number
        cin.clear();
        cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');

        cout << "Invalid Number, Enter a valid one:" << endl;

        cin >> Number;
    }

    return Number;
}

int main()
{
    cout << "Your Number is:" << ReadNumber();

}
```



Lesson #12 Code

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

int main()
{
    cout << "Result:" << (12 & 25);
}
```



Lesson #13 Code

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

int main()
{
    cout << "Result:" << (12 | 25);
}
```



Lesson #14 Code

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

// function declaration
void add(int, int);

int main() {
    add(10, 20);
    return 0;
}

// function Definition

void add(int a, int b) {
    cout << (a + b);
}
```



Lesson #15 Code

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

int MySum(int a, int b, int c = 0, int d = 0) {
    return (a + b + c + d);
}

int main() {

    cout << MySum(10, 20) << endl;
    cout << MySum(10, 20, 30) << endl;
    cout << MySum(10, 20, 30, 40) << endl;

    return 0;
}
```



Lesson #16 Code – No Overloading

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

double MySum2DoubleNumbers(double a, double b) {
    return (a + b);
}

int MySum2IntegerNumbers(int a, int b) {
    return (a + b);
}

int MySum3IntegerNumbers(int a, int b, int c) {
    return (a + b + c);
}

int MySum4IntegerNumbers(int a, int b, int c, int d) {
    return (a + b + c + d);
}

int main() {

    cout << MySum2IntegerNumbers(10, 20) << endl;
    cout << MySum2DoubleNumbers(10.1, 20.1) << endl;
    cout << MySum3IntegerNumbers(10, 20, 30) << endl;
    cout << MySum4IntegerNumbers(10, 20, 30, 40) << endl;

    return 0;
}
```



Lesson #17 Code

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

void Function4()
{
    cout << "Hi I'm function4 " << endl;
}

void Function3()
{
    Function4();
}

void Function2()
{
    Function3();
}

void Function1()
{
    Function2();
}

int main() {
    Function1();
    return 0;
}
```



Lesson #18 Code

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

void Function2();

void Function4()
{
    cout << "Hi I'm function4 " << endl;
}

void Function3()
{
    Function4();
}

void Function2()
{
    Function3();
}

void Function1()
{
    Function2();
    Function4();
}

int main() {
    Function1();
    return 0;
}
```



Lesson #19 Code

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

void PrintNumbersFromNtoM(int N, int M)
{
    if (N <= M)
    {
        cout << N << endl;
        PrintNumbersFromNtoM(N + 1, M);
    }
}

int main() {
    PrintNumbersFromNtoM(1, 10);
    return 0;
}
```



Lesson #19 Homework1 Solution

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

void PrintNumbersFromMtoN(int M, int N)
{
    if (M >= N)
    {
        cout << M << endl;
        PrintNumbersFromMtoN(M - 1, N);
    }
}

int main()
{
    PrintNumbersFromMtoN(10, 1);
    return 0;
}
```



Lesson #19 Homework2 Solution

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

int MyPower(int Base, int Power)
{
    if (Power == 0)
        return 1;
    else
    {
        return (Base * MyPower(Base, Power - 1));
    }
}

int main() {
    cout << MyPower(2, 4);

    return 0;
}
```



Lesson #20 Static Variables

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

void MyFunc() {
    static int Number = 1;
    cout << "Value of Number: " << Number << "\n";
    Number++;
}

int main()
{
    MyFunc();
    MyFunc();
    MyFunc();

    return 0;
}
```



Lesson #21 Auto Variables

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

int main()
{
    auto x = 10; //Type Integer
    auto y = 12.5; //Type Double
    auto z = "Mohammed Abu-Hadoud"; //Type String

    cout << x << endl;
    cout << y << endl;
    cout << z << endl;

    return 0;
}
```



Lesson #23 Integer Format (printf)

```
#include <iostream>
#include <cstdio>

using namespace std;

int main() {

    int Page = 1, TotalPages = 10;

    // print string and int variable
    printf("The page number = %d \n", Page);
    printf("You are in Page %d of %d \n", Page, TotalPages);

    //Width specification
    printf("The page number = %0*d \n", 2, Page);
    printf("The page number = %0*d \n", 3, Page);
    printf("The page number = %0*d \n", 4, Page);
    printf("The page number = %0*d \n", 5, Page);

    int Number1 = 20, Number2 = 30;
    printf("The Result of %d + %d = %d \n", Number1, Number2,
Number1+ Number2);

    return 0;
}
```



Lesson #24 Float Format (printf)

```
#include <iostream>
#include <cstdio>

using namespace std;

int main() {

    float PI = 3.14159265;

    //Precision specification
    printf("Precision specification of %.1f\n", 1, PI);
    printf("Precision specification of %.2f\n", 2, PI);
    printf("Precision specification of %.3f\n", 3, PI);
    printf("Precision specification of %.4f\n", 4, PI);

    float x = 7.0, y = 9.0;
    printf("\nThe float division is : %.3f / %.3f = %.3f \n\n", x,
y, x / y);

    double d = 12.45;
    printf("The double value is : %.3f \n", d);
    printf("The double value is : %.4f \n", d);

    return 0;
}
```



Lesson #25 String and Char Format (printf)

```
#include <iostream>
#include <cstdio>

using namespace std;

int main() {

    char Name[] = "Mohammed Abu-Hadhoud";
    char SchoolName[] = "Programming Advices";

    // print string and String
    printf("Dear %s, How are you?\n\n", Name);
    printf("Welcome to %s School!\n\n", SchoolName);

    char c = 'S';

    printf("Setting the width of c :%*c \n", 1, c);
    printf("Setting the width of c :%*c \n", 2, c);
    printf("Setting the width of c :%*c \n", 3, c);
    printf("Setting the width of c :%*c \n", 4, c);
    printf("Setting the width of c :%*c \n", 5, c);

    return 0;
}
```



Lesson #26 – Setw Manipulator

```
#include <iostream>
#include <iomanip> // this library stored the std::setw
using namespace std;

int main()
{
    cout << "-----|-----|-----|-----"
        | " << endl;

    cout << "  Code   |           Name          |  Mark
        | " << endl;

    cout << "-----|-----|-----|-----"
        | " << endl;

    cout << setw(9) << "C101" << "|" << setw(32) << "introduction
to Programming 1" << "|" << setw(9) << "95" << "|" << endl;

    cout << setw(9) << "C102" << "|" << setw(32) << "Computer
Hardware" << "|" << setw(9) << "88" << "|" << endl;

    cout << setw(9) << "C1035243" << "|" << setw(32) << "Network"
<< "|" << setw(9) << "75" << "|" << endl;

    cout << "-----|-----|-----|-----"
        | " << endl;

    return 0;
}
```



Lesson #27 - Code

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

int main()
{
    //int x[Rows][Cols];
    int x[3][4] = { {1,2,3,4},
                    {5,6,7,8},
                    {9,10,11,12}
                };

    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            cout << x[i][j] << " ";
        }

        cout << endl;
    }

    return 0;
}
```



Lesson #27 - Homework Solution

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

int main()
{
    //int x[Rows][Cols];
    int x[10][10];

    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            x[i][j] = (i + 1) * (j + 1);
        }
    }

    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            printf("%0*d ", 2, x[i][j]);
        }

        cout << endl;
    }

    return 0;
}
```



Lesson #28 - Code

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

int main()
{
    // std::vector<T> vector_name;

    vector <int> vNumbers = { 10, 20, 30, 40, 50 };

    cout << "Numbers Vector = ";

    // ranged loop
    for (int &Number : vNumbers) {

        cout << Number << " ";
    }

    cout << endl;

    return 0;
}
```



Lesson #29 - Code

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

int main()
{
    vector <int> vNumbers;

    vNumbers.push_back(10);
    vNumbers.push_back(20);
    vNumbers.push_back(30);
    vNumbers.push_back(40);
    vNumbers.push_back(50);

    cout << "Numbers Vector: \n\n";

    // ranged loop
    for (int & Number : vNumbers) {
        cout << Number << endl;
    }

    cout << endl;

    return 0;
}
```



Lesson #29- Homework Solution

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

void ReadNumbers(vector <int> & vNumbers)
{
    char ReadMore = 'Y';
    int Number;

    while (ReadMore == 'Y' || ReadMore == 'y')
    {
        cout << "Please enter a number? ";
        cin >> Number;

        vNumbers.push_back(Number);

        cout << "\nDo you want to read more numbers? Y/N ?";
        cin >> ReadMore;
    }
}

void PrintVectorNumbers(vector <int> & vNumbers)
{
    cout << "Numbers Vector: \n\n";

    // ranged loop
    for (int Number : vNumbers)
    {
        cout << Number << endl;
    }

    cout << endl;
}

int main()
{
    vector <int> vNumbers;

    ReadNumbers(vNumbers);
    PrintVectorNumbers(vNumbers);

    return 0;
}
```



Lesson #30 - Code

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

struct stEmployee
{
    string FirstName;
    string LastName;
    float Salary;
};

int main()
{
    // std::vector<T> vector_name;
    vector <stEmployee> vEmployees;

    stEmployee tempEmployee;

    tempEmployee.FirstName = "Mohammed";
    tempEmployee.LastName = "Abu-Hahdoud";
    tempEmployee.Salary = 5000;
    vEmployees.push_back(tempEmployee);

    tempEmployee.FirstName = "Ali";
    tempEmployee.LastName = "Maher";
    tempEmployee.Salary = 300;
    vEmployees.push_back(tempEmployee);

    tempEmployee.FirstName = "Aya";
    tempEmployee.LastName = "Omran";
    tempEmployee.Salary = 1000;
    vEmployees.push_back(tempEmployee);

    cout << "Employees Vector: \n\n";

    for (stEmployee &Employee : vEmployees) {

        cout << "FirstName: " << Employee.FirstName << endl;
        cout << "LastName : " << Employee.LastName << endl;
        cout << "Salary    : " << Employee.Salary << endl;
        cout << endl;
    }

    cout << endl;
    return 0;
}
```



Lesson #30- Homework Solution

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

struct stEmployee
{
    string FirstName;
    string LastName;
    float Salary;

};

void ReadEmployees(vector <stEmployee>& vEmployees)
{
    char ReadMore = 'Y';
    stEmployee tempEmployee;

    while (ReadMore == 'Y' || ReadMore == 'y')
    {
        cout << "Enter FirstName? ";
        cin >> tempEmployee.FirstName;

        cout << "Enter LastName? ";
        cin >> tempEmployee.LastName;

        cout << "Enter Salary? ";
        cin >> tempEmployee.Salary;

        vEmployees.push_back(tempEmployee);

        cout << "\nDo you want to read more employees? Y/N ?";
        cin >> ReadMore;
    }
}
```



Lesson #30- Homework Solution

```
void PrintEmployees(vector <stEmployee>& vEmployees)
{
    cout << "\nEmployees Vector: \n\n";

    // ranged loop
    for (stEmployee& Employee : vEmployees) {

        cout << "FirstName: " << Employee.FirstName << endl;
        cout << "LastName : " << Employee.LastName << endl;
        cout << "Salary   : " << Employee.Salary << endl;
        cout << endl;
    }

    cout << endl;
}

int main()
{

    // std::vector<T> vector_name;
    vector <stEmployee> vEmployees;

    ReadEmployees(vEmployees);
    PrintEmployees(vEmployees);

    return 0;
}
```



Lesson #31 - Code

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

int main()
{
    vector <int> vNumbers;

    vNumbers.push_back(10);
    vNumbers.push_back(20);
    vNumbers.push_back(30);
    vNumbers.push_back(40);
    vNumbers.push_back(50);

    vNumbers.pop_back();
    vNumbers.pop_back();
    vNumbers.pop_back();
    vNumbers.pop_back();
    vNumbers.pop_back();

    cout << "Numbers Vector: \n\n";

    // ranged loop
    for (int & Number : vNumbers) {
        cout << Number << endl;
    }

    cout << endl;

    return 0;
}
```



Lesson #32 - Code

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

int main()
{
    vector <int> vNumbers;

    vNumbers.push_back(10);
    vNumbers.push_back(20);
    vNumbers.push_back(30);
    vNumbers.push_back(40);
    vNumbers.push_back(50);

    cout << "First Element: " << vNumbers.front() << endl;
    cout << "Last Element: " << vNumbers.back() << endl;

    //returns the number of elements present in the vector
    cout << "Size: " << vNumbers.size() << endl;

    //check the overall size of a vector
    cout << "Capacity : " << vNumbers.capacity() << endl;

    //returns 1 (true) if the vector is empty
    cout << "Empty : " << vNumbers.empty() << endl;

    return 0;
}
```



Lesson #33 - ByRef

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

void Function1(int &x)
{
    x++;
}

int main()
{
    int a = 10;

    Function1(a);

    cout << "\n a after calling function1 = " << a << endl;

    return 0;
}
```



Lesson #33 - ByVal

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

void Function1(int x)
{
    x++;
}

int main()
{
    int a = 10;

    Function1(a);

    cout << "\n a after calling function1 = " << a << endl;

    return 0;
}
```



Lesson #33 - Code 3

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

int main()
{
    int a = 10;

    cout << "Value of           a = " << a << endl;
    cout << "Address/Reference of a = " << &a << endl;

    return 0;
}
```



Lesson #34 - Code

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

int main()
{
    int a = 10;
    int & x = a;

    cout << &a << endl;
    cout << &x << endl;

    cout << a << endl;
    cout << x << endl;

    x = 20;

    cout << a << endl;
    cout << x << endl;

    a = 30;

    cout << a << endl;
    cout << x << endl;

    return 0;
}
```



Lesson #35 - Code

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

int main()
{
    int a = 10;

    cout <<"a value      = " << a << endl;
    cout << "a address   = "<< &a << endl;

    int * p;
    p = &a;

    cout << "Pointer Value = " << p;

    cout << endl;

    return 0;
}
```



Lesson #36 - Code

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

int main()
{
    int a = 10;

    cout << "a value      = " << a << endl;
    cout << "a address    = "<< &a << endl;

    int * p;
    p = &a;

    cout << "Pointer Value = " << p << endl;
    cout << "Value of the address that p is pointing to is " << *p << endl;

    *p = 20;

    cout << a << endl;
    cout << *p << endl;

    cout << endl;

    return 0;
}
```



Lesson #37 - Code

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

int main()
{
    int x, * p;

    // Wrong!
    // p is an address but x is not
    p = x;

    // Correct!
    // p is an address and so is &x
    p = &x;

    // Wrong!
    // &x is an address
    // *p is the value stored in &x
    *p = &x;

    // Correct!
    // *p is the value and so x
    *p = x;

    return 0;
}
```



Lesson #37 - Code

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

int main()
{
    int x, * p;

    // Wrong!
    // p is an address but x is not
    p = x;

    // Correct!
    // p is an address and so is &x
    p = &x;

    // Wrong!
    // &x is an address
    // *p is the value stored in &x
    *p = &x;

    // Correct!
    // *p is the value and so x
    *p = x;

    return 0;
}
```



Lesson #39 – Swap Using Pointers

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

void swap(int *n1, int *n2)
{
    int temp;
    temp = *n1;
    *n1 = *n2;
    *n2 = temp;
}

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;
}
```



Lesson #38 - Code

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

int main()
{
    int a = 10;
    int& x = a;

    cout << &a << endl;
    cout << &x << endl;

    cout << a << endl;
    cout << x << endl;

    int* p = &a;

    cout << p << endl;
    cout << *p << endl;

    int b = 20;
    p = &b;

    cout << p << endl;
    cout << *p << endl;

    return 0;
}
```



Lesson #39 – Swap Using Pointers

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

void swap(int *n1, int *n2)
{
    int temp;
    temp = *n1;
    *n1 = *n2;
    *n2 = temp;
}

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;
}
```



Lesson #39 – Swap Using Reference

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

void swap(int &n1, int &n2)
{
    int temp;
    temp = n1;
    n1 = n2;
    n2 = temp;
}

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;
}
```



Lesson #40 – Code

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

int main()
{
    int arr[4] = {10,20,30,40};
    int *ptr;
    ptr = arr;

    //ptr is equivalent to &arr[0];
    //ptr + 1 is equivalent to &arr[1];
    //ptr + 2 is equivalent to &arr[2];
    //ptr + 3 is equivalent to &arr[3];

    cout << "Addresses are:\n";

    cout << ptr << endl;
    cout << ptr + 1 << endl;
    cout << ptr + 2 << endl;
    cout << ptr + 3 << endl;

    cout << "\nValues are: \n";
    cout << *(ptr) << endl;
    cout << *(ptr + 1) << endl;
    cout << *(ptr + 2) << endl;
    cout << *(ptr + 3) << endl;

    return 0;
}
```



Lesson #41 – Code

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

struct stEmployee
{
    string Name;
    float Salary;
};

int main()
{
    stEmployee Employee1, *ptr;

    Employee1.Name = "Mohammed Abu-Hadhoud";
    Employee1.Salary = 5000;

    cout << Employee1.Name << endl;
    cout << Employee1.Salary << endl;

    ptr = &Employee1;

    cout << "\nUsing Pointer:\n";
    cout << ptr->Name << endl;
    cout << ptr->Salary << endl;

    return 0;
}
```



Lesson #42 – Code

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

int main()
{
    void * ptr;

    float f1 = 10.5;
    int x = 50;

    ptr = &f1;

    cout << ptr << endl;
    cout << *(static_cast<float*>(ptr)) << endl;

    ptr = &x;

    cout << ptr << endl;
    cout << *(static_cast<int*>(ptr)) << endl;

    return 0;
}
```



Lesson #43 – Code

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

int main()
{
    // declare an int pointer
    int* ptrX;

    // declare a float pointer
    float* ptrY;

    // dynamically allocate memory
    ptrX = new int;
    ptrY = new float;

    // assigning value to the memory
    *ptrX = 45;
    *ptrY = 58.35f;

    cout << *ptrX << endl;
    cout << *ptrY << endl;

    // deallocate the memory
    delete ptrX;
    delete ptrY;

    return 0;
}
```



Lesson #44 – Code

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

int main() {

    int num;
    cout << "Enter total number of students: ";
    cin >> num;

    float * ptr;

    // memory allocation of num number of floats
    ptr = new float[num];

    cout << "Enter grades of students." << endl;

    for (int i = 0; i < num; i++) {
        cout << "Student" << i + 1 << ": ";
        cin >> *(ptr + i);
    }

    cout << "\nDisplaying grades of students." << endl;
    for (int i = 0; i < num; i++) {
        cout << "Student" << i + 1 << ": " << *(ptr + i) << endl;
    }

    // ptr memory is released
    delete[] ptr;

    return 0;
}
```



Lesson #46 – Code

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

int main() {

    vector<int> num{ 1, 2, 3, 4, 5 };

    cout << "\n\n using .at(i) \n";
    cout << "Element at Index 0: " << num.at(0) << endl;
    cout << "Element at Index 2: " << num.at(2) << endl;
    cout << "Element at Index 4: " << num.at(4) << endl;

    cout << "\n\n using [i]\n";
    cout << "Element at Index 0: " << num[0] << endl;
    cout << "Element at Index 2: " << num[2] << endl;
    cout << "Element at Index 4: " << num[5] << endl;

    return 0;
}
```



Lesson #47 – Code

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

int main() {

    vector<int> num{ 1, 2, 3, 4, 5 };

    cout << "Initial Vector: ";

    for (const int& i : num) {
        cout << i << " ";
    }

    cout << "\n\nUpdated Vector: ";

    for ( int& i : num) {

        i = 20;
        cout << i << " ";
    }

    num[1] = 40;
    num.at(2) = 80;
    num.at(4) = 90;

    cout << "\n\nUpdated Vector: ";

    for (const int& i : num) {
        cout << i << " ";
    }

    return 0;
}
```



Lesson #47 – Code

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

int main() {

    vector<int> num{ 1, 2, 3, 4, 5 };

    // declare iterator
    vector<int>::iterator iter;

    // use iterator with for loop
    for (iter = num.begin(); iter != num.end(); iter++)
    {

        cout << *iter << "   ";

    }

    return 0;
}
```



Lesson #49 – Code

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

int main() {

    vector<int> num{ 1, 2, 3, 4, 5 };

    try
    {
        cout << num.at(5);
    }
    catch (...)
    {

        cout << "out of bound\n";
    }

    return 0;
}
```



Lesson #50 – Code

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    string S1= "My Name is Mohammed Abu-Hadoud, I Love
Programming.";

    //Prints the length of the string
    cout << S1.length() << endl;

    //Returns the letter at position 3
    cout << S1.at(3) << endl;

    //Adds @ProgrammingAdvices to the end of string
    S1.append(" @ProgrammingAdvices");
    cout << S1 << endl;

    //inserts Ali at position 7
    S1.insert(7, " Ali ");
    cout << S1 << endl;

    //Prints all the next 8 letters from position 16.
    cout << S1.substr(16,8) << endl;

    //Adds one character to the end of the string
    S1.push_back('X');
    cout << S1 << endl;

    //Removes one character from the end of the string
    S1.pop_back();
    cout << S1 << endl;

    //Finds Ali in the string
    cout << S1.find("Ali") << endl;
    //Finds ali in the string
    cout << S1.find("ali") << endl;

    if (S1.find("ali") == S1npos)
    {
        cout << "ali is not found";
    }

    //clears all string letters.
    S1.clear();
    cout << S1 << endl;
    return 0;
}
```



Lesson #51 – Code

```
#include <iostream>
#include <string>
#include <cctype>

using namespace std;

int main()
{
    char x ;
    char w ;

    x = toupper('a');
    w = tolower('A');

    cout << "converting a to A: " << x << endl;
    cout << "converting A to a: " << w << endl;

    // Digits (A to Z)
    // returns zero if not, and non zero of yes
    cout << "isupper('A') " << isupper( 'A') << endl;

    // lower case (a to z)
    // returns zero if not, and non zero of yes
    cout << "islower('A') " << islower('A') << endl;

    // Digits (0 to 9)
    // returns zero if not, and non zero of yes
    cout << "isdigit('A') " << isdigit ('A') << endl;

    // punctuation characters are !#$%&'()*/,-./:;<=>?@[\\]^_`{|}~
    // returns zero if not, and non zero of yes
    cout << "ispunct('A') " << ispunct('A') << endl;

    return 0;
}
```



Lesson #52 – Code

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

int main() {

    fstream MyFile;

    MyFile.open("MyFile.txt", ios::out); //Write Mode

    if (MyFile.is_open())
    {

        MyFile << "Hi, this is the first line\n";
        MyFile << "Hi, this is the second line\n";
        MyFile << "Hi, this is the third line\n";

        MyFile.close();
    }

    return 0;
}
```



Lesson #53 – Code

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

int main() {

    fstream MyFile;
    MyFile.open("MyFile.txt",  ios::out | ios::app );//append Mode

    if (MyFile.is_open())
    {

        MyFile << "Hi, this is a new line\n";
        MyFile << "Hi, this is another new line\n";
        MyFile.close();

    }

    return 0;
}
```



Lesson #54 – Code

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

void PrintFileContent(string FileName)
{
    fstream MyFile;
    MyFile.open(FileName, ios::in); //read Mode

    if (MyFile.is_open())
    {

        string Line;

        while (getline(MyFile, Line))
        {

            cout << Line << endl;
        }

        MyFile.close();
    }
}

int main() {
    PrintFileContent("MyFile.txt");
    return 0;
}
```



Lesson #55 – Code

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>

using namespace std;

void LoadDataFromFileToVector(string FileName, vector <string>& vFileContent)
{
    fstream MyFile;
    MyFile.open(FileName, ios::in); //read Mode

    if (MyFile.is_open())
    {

        string Line;

        while (getline(MyFile, Line))
        {
            vFileContent.push_back(Line);
        }

        MyFile.close();
    }
}

int main() {

    vector <string> vFileContent;

    LoadDataFromFileToVector("MyFile.txt", vFileContent);

    for (string &Line : vFileContent)
    {
        cout << Line << endl;
    }

    return 0;
}
```



Lesson #56 – Code

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>

using namespace std;

void SaveVectorToFile(string FileName, vector <string>
vFileContent)
{
    fstream MyFile;
    MyFile.open("MyFile.txt", ios::out);

    if (MyFile.is_open())
    {

        for (string &Line : vFileContent)
        {
            if (Line != "")
            {
                MyFile << Line << endl;
            }
        }

        MyFile.close();
    }
}

int main()
{

    vector <string> vFileContent{
"Ali", "Shadi", "Maher", "Fadi", "Lama"};

    SaveVectorToFile("MyFile.txt", vFileContent);

    return 0;
}
```



Lesson #57 – Code

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>

using namespace std;

void LoadDataFromFileToVector(string FileName, vector <string>& vFileContent)
{
    fstream MyFile;
    MyFile.open("MyFile.txt", ios::in); //read Mode

    if (MyFile.is_open())
    {
        string Line;
        while (getline(MyFile, Line))
        {
            vFileContent.push_back(Line);
        }

        MyFile.close();
    }
}

void SaveVectorToFile(string FileName, vector <string> vFileContent)
{
    fstream MyFile;
    MyFile.open("MyFile.txt", ios::out);

    if (MyFile.is_open())
    {

        for (string Line : vFileContent)
        {
            if (Line != "")
            {
                MyFile << Line << endl;
            }
        }

        MyFile.close();
    }
}
```



Lesson #57 – Code

```
void DeleteRecordFromFile(string FileName, string Record)
{
    vector <string> vFileContent;
    LoadDataFromFileToVector(FileName, vFileContent);

    for (string &Line : vFileContent)
    {
        if (Line == Record)
        {
            Line = "";
        }
    }

    SaveVectorToFile(FileName, vFileContent);
}

void PrintFileContent(string FileName)
{
    fstream MyFile;
    MyFile.open(FileName, ios::in); //read Mode

    if (MyFile.is_open())
    {

        string Line;

        while (getline(MyFile, Line))
        {
            cout << Line << endl;
        }
        MyFile.close();
    }
}

int main() {

    cout << "File Content Before Delete.\n";
    PrintFileContent("MyFile.txt");

    DeleteRecordFromFile("MyFile.txt", "Ali");

    cout << "\n\nFile Content After Delete.\n";
    PrintFileContent("MyFile.txt");

    return 0;
}
```



Lesson #58 – Code

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>

using namespace std;

void LoadDataFromFileToVector(string FileName, vector <string>& vFileContent)
{
    fstream MyFile;
    MyFile.open("MyFile.txt", ios::in); //read Mode

    if (MyFile.is_open())
    {
        string Line;
        while (getline(MyFile, Line))
        {
            vFileContent.push_back(Line);
        }

        MyFile.close();
    }
}

void SaveVectorToFile(string FileName, vector <string> vFileContent)
{
    fstream MyFile;
    MyFile.open("MyFile.txt", ios::out);

    if (MyFile.is_open())
    {
        for (string Line : vFileContent)
        {
            if (Line != "")
            {
                MyFile << Line << endl;
            }
        }

        MyFile.close();
    }
}
```



Lesson #58 – Code

```
void UpdateRecordInFile(string FileName, string Record, string
UpdateTo)
{
    vector <string> vFileContent;
    LoadDataFromFileToVector(FileName, vFileContent);

    for (string &Line : vFileContent)
    {
        if (Line == Record)
        {
            Line = UpdateTo;
        }

    }
    SaveVectorToFile(FileName, vFileContent);
}

void PrintFileContent(string FileName)
{
    fstream MyFile;
    MyFile.open(FileName, ios::in); //read Mode

    if (MyFile.is_open())
    {

        string Line;

        while (getline(MyFile, Line))
        {
            cout << Line << endl;
        }

        MyFile.close();
    }

}
```



Lesson #58 – Code

```
int main() {  
  
    cout << "File Content Before Delete.\n";  
    PrintFileContent("MyFile.txt");  
  
    UpdateRecordInFile("MyFile.txt", "Ali", "Omar");  
  
    cout << "\n\nFile Content After Delete.\n";  
    PrintFileContent("MyFile.txt");  
  
    return 0;  
}
```



Lesson #59 – Code

```
#pragma warning(disable : 4996)

#include <ctime>
#include <iostream>

using namespace std;

int main()
{
    time_t t = time(0);      // get time now

    char* dt = ctime(&t);   // convert in string form
    cout << "Local date and time is: " << dt << "\n";

    // converting now to tm struct for UTC date/time
    tm* gmtm = gmtime(&t);

    dt = asctime(gmtm);
    cout << "UTC date and time is: " << dt;

}
```



Lesson #60 – Code

```
#pragma warning(disable : 4996)

#include <ctime>
#include <iostream>

using namespace std;

/*
    int tm_sec; // seconds of minutes from 0 to 61
    int tm_min; // minutes of hour from 0 to 59
    int tm_hour; // hours of day from 0 to 24
    int tm_mday; // day of month from 1 to 31
    int tm_mon; // month of year from 0 to 11
    int tm_year; // year since 1900
    int tm_wday; // days since sunday
    int tm_yday; // days since January 1st
    int tm_isdst; // hours of daylight savings time
*/

int main() {
    time_t t = time(0);      // get time now

    tm* now = localtime(&t);

    cout << "Year: " << now->tm_year + 1900 << endl;
    cout << "Month: " << now->tm_mon + 1 << endl;
    cout << "Day: " << now->tm_mday << endl;
    cout << "Hour: " << now->tm_hour << endl;
    cout << "Min: " << now->tm_min << endl;
    cout << "Second: " << now->tm_sec << endl;
    cout << "Week Day (Days since sunday): " << now->tm_wday << endl;
    cout << "Year Day (Days since Jan 1st): " << now->tm_yday << endl;
    cout << "hours of daylight savings:" << now->tm_isdst << endl;

}
```