

## Reading and writing to files

## Reading and writing to Data Files

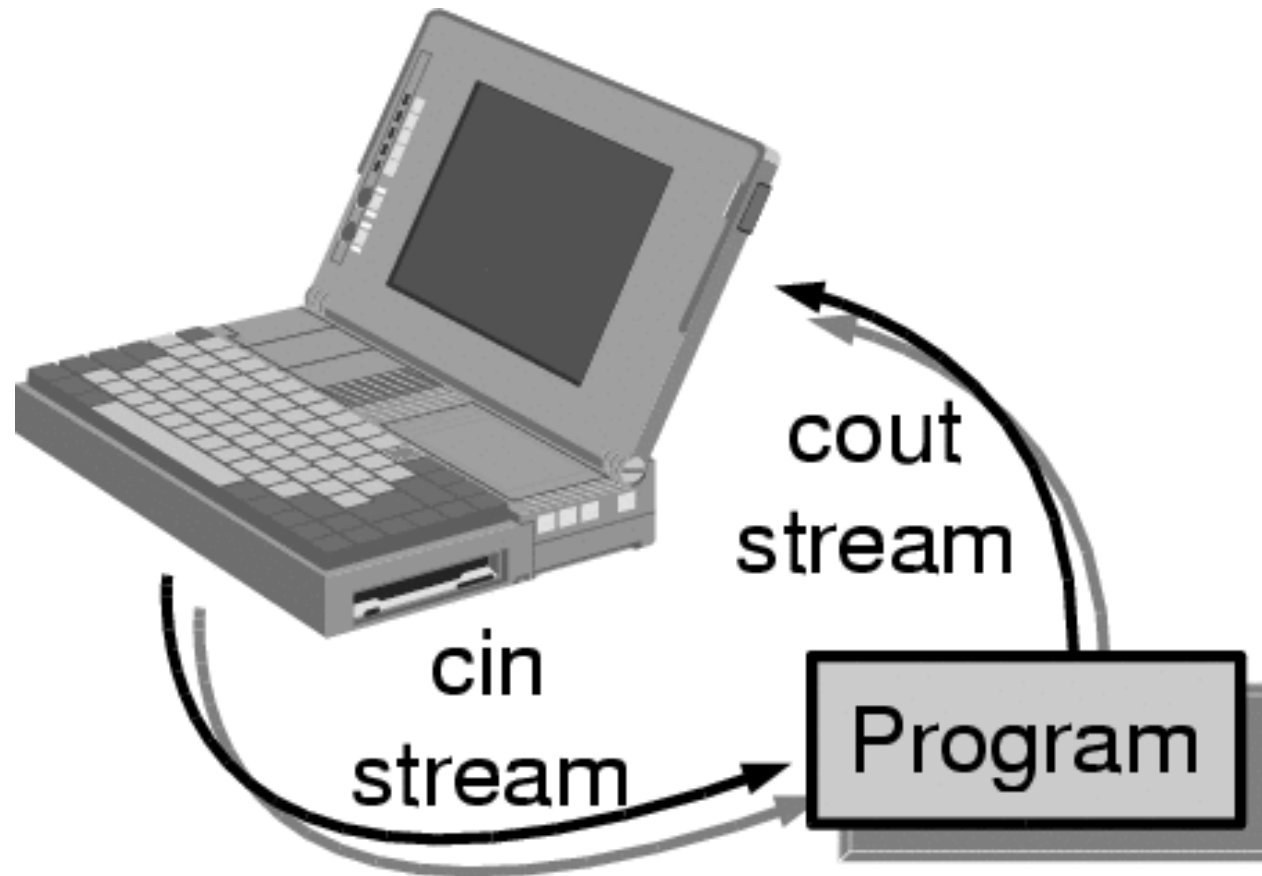
### **ifstream and ofstream...streams:**

1. Used to read from keyboard, or write to screen
2. Can connect to and from files
  - (a) ifstream: read from file (i=input)
  - (b) ofstream: write to file (o=output)
3. Works just like any other stream, except:
  - (a) Needs to be attached to file
  - (b) Declaration of data object of type **ofstream** or **ifstream**
  - (c) Then feed information into stream like use of **cin** or **cout**

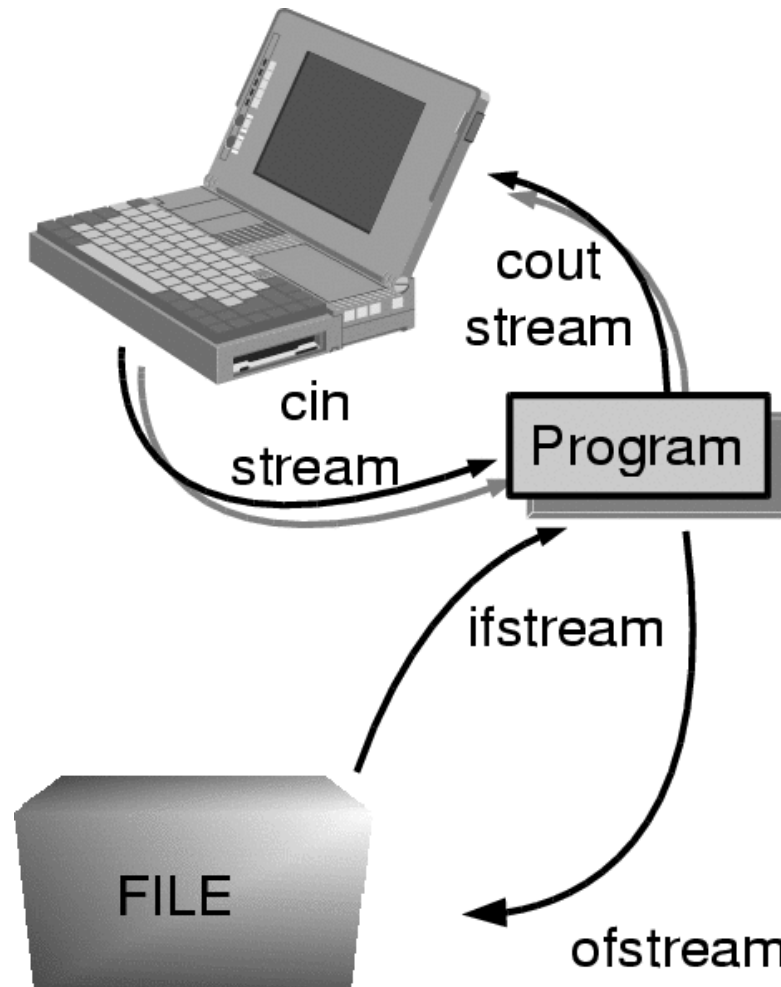
# CS149: Elements of Computer Science

---

## Streams



## File input and output streams



## ofstream example

```
1 // test ofstreams
2 #include <string>
3 #include <iostream.h>
4 #include <fstream.h>
5
6 int main (void){
7
8     int i;
9     double r,d;
10
11     ofstream fout ("myfile.txt");
12
13     r = 0;
14     for (i=0; i<10; i++){
15         d = 6.28 * r;
16         fout << r << "\t" << d << endl;
17         r++;
18     }
```

# CS149: Elements of Computer Science

---

```
19     return 0;  
20 }
```

## ifstream example

```
1 // test ifstreams
2 #include<fstream.h>
3 #include<iostream.h>
4
5 int main (void){
6
7     int i;
8     double r,d;
9     ifstream fin("anotherfile.dat");
10
11     for (i=0; i<10; i++){
12         fin >> r >> d;
13         cout << i << ")_ " << r << "\t" << d << endl;
14     }
15
16     fin.close();
17     return 0;
18 }
```

## more ifstreams

```
1 // ifstream_manip.cpp: reads a file and calculates
2 #include<fstream.h>
3 #include<iostream.h>
4
5 int main (void){
6
7     int i=1;
8     double r,d;
9     ifstream fin ("anothertest.dat");
10    ofstream fout ("our_output.dat");
11
12    while (fin >> r){
13        d = r * 3.1415;
14        fout << r << "\t" << d << endl;
15        i++;
16    }
17    return 0;
18 }
```

# CS149: Elements of Computer Science

---

## Opening and closing file streams

```
1  # include <iostream.h>
2  # include <fstream.h>
3  # include <string>
4
5  int main (void){
6
7      int a, b, c;
8      string answer;
9      ifstream fin;
10
11     cout << "Open_file_first_time!\n";
12     // open file and read three values
13     fin.open ("myfile.txt", ios::in);
14     fin >> a >> b >> c;
15     cout << a << ", " << b << ", " << c << endl;
16     fin.close();
17
18     // halt program for a sec
```

# CS149: Elements of Computer Science

---

```
19     cout << "Press any key and enter to continue\n";
20     cin >> answer;
21
22     cout << "Open file second time!\n";
23     // and again
24     fin.open ("myfile.txt", ios::in);
25     fin >> a >> b >> c;
26     cout << a << ", " << b << ", " << c << endl;
27     fin.close();
28
29     return 0;
30
31 }
```

# CS149: Elements of Computer Science

---

Now on to other matters: arrays.

1. All program up to this point: 1 variable for 1 value
2. This means: 100 values to store = 100 variable names
3. Solution:
  - (a) Arrays: index a given variable
  - (b) Use variable name and index to store values
  - (c) Convenient access by index
  - (d) Declaration determine variable name and number of stored values
4. Allows one variable, so to speak, to store set of values, each indexed by specific index value
5. Variable types also apply to arrays

## Arrays Declaration and Indexes

### Three Components:

1. Type: int, double, etc...
2. Array name
3. Size: size = integer constant expresses number of values

### Example:

1. **int a[4]**
2. **double z[5]**

### Indexing:

1. First Index = 0
2. Last index = size - 1
3. **NO BOUNDARY CHECKING!**

Operates like other variables, except indexed.Examples:

1. **a[3] = 1**
2. **a[0] = 2**

## Arrays Example

```
1 // arrays_declaration.cpp: example of array declaration
2 #include<iostream.h>
3
4 int main (void){
5
6     // a is int array
7     int a[3], j, x;
8
9     for (j=0; j<3; j++){
10         a[j] = j*j;
11     }
12
13     cout << "index 0:" << a[0];
14     cout << "\tindex 1:" << a[1];
15     cout << "\tindex 2:" << a[2] << endl;
16
17     return 0;
18 }
```

## Arrays: Underlying Representation

### One-Dimensional Arrays:

1. Example: `int a[5]`
2. Variable **a** identifies memory range suited to store 5 integers
3. Range starts at memory location for **a**
4. First item: `a+0`, therefore first item ~ index 0
5. Last item: `a+4`

**int a[5];**

|   |      |      |      |      |      |
|---|------|------|------|------|------|
| a | -    | -    | -    | -    | -    |
|   | a[0] | a[1] | a[2] | a[3] | a[4] |

**a[2] = 5;**

|   |      |      |      |      |      |
|---|------|------|------|------|------|
| a | -    | -    | 5    | -    | -    |
|   | a[0] | a[1] | a[2] | a[3] | a[4] |

**a[10] = 2;??**

## Arrays: Index boundary checking.

1. Bad news: there is none
2. All index values are allowed
3. The consequences of over or underindexing can be severe:
  - (a) Array indexes identify memory locations
  - (b) Index beyond allotted number in variable declaration may identify valid memory location
  - (c) Except, it is one in use by other processes
  - (d) Consequence: program or **computer** crashes

# CS149: Elements of Computer Science

---

Arrays: Index boundary checking.

```
1 // arrays_boundaries.cpp: example of array declaration
2 #include<iostream.h>
3
4 int main (void){
5     // a is int array
6     int a[3],j;
7
8     for (j=0; j<10; j++){
9         a[j] = j*j;
10    }
11
12    for (j=0; j<10; j++){
13        cout << j << " :_ " << a[j] << endl;
14    }
15
16    return 0;
17 }
```

## Arrays: Initialization.

1. When variables are declared they can be assigned initial values
  - (a) Example, **int a = 4;**
  - (b) Can be changed later
2. Same applies to arrays:
  - (a) Syntax: **type name = {type constants, ...}**
  - (b) Applies to **char** type as well: strings!

# CS149: Elements of Computer Science

---

## Arrays: Initialization.

```
1 // arrays_initialization.cpp: example of array declaration
2 #include<iostream.h>
3
4 int main (void){
5     // a is int array
6     int a[3] = {5, 13, 21}, j;
7     int b[] = {6, 7, 8, 9};
8     double c[8] = {50};
9     char alphabet[] = {'a', 'b', 'c', 'd', 'e',
10                        'f', 'g', 'h', 'i', 'j', 'z'};
11     char word[] = "A_word";
12
13     for (j=0; j<6; j++){
14         cout << j << " :_" << word[j] << endl;
15     }
16
17     return 0;
18 }
```

## Arrays: Other data objects

```
1  # include <iostream.h>
2  # include <string>
3
4  int main (void){
5
6      int i;
7      string words[5] = { "C++" , "is" , "so" , "much" , "fun!" };
8
9      for (i=0; i<5; i++){
10         cout << words[i] << endl;
11     }
12
13     return 0;
14
15 }
```

# CS149: Elements of Computer Science

---

Arrays: Let's try to find maximum value