CS 101: Computer Programming and Utilization

Shivaram Kalyanakrishnan
(Abhiram Ranade's slides, borrowed and edited)
Lecture 14

Today's Lecture

- Creating arrays, accessing elements
- Example program
- Tool: input-output redirection

Computers must deal with large amounts of data

- Simulate what happens when many balls are moving in a box. (Gas molecules?)
- Given altitudes of various points in a lake, find how much water is there given the water level.
- Given the road map of India, find the shortest route from Nagpur to Pudukottai.

How to handle lot of data?

- Fundamental problem: Writing out variable names to store information would be tiring

```
double pressure1, pressure2, …,
pressure1000;
```

- This is the problem solved using Arrays.
- More elaborate, modern, and flexible solution involving “vector”s will be discussed later if we have time.
- Arrays are simple to understand. Ideas useful in vectors too.

Arrays

```
double pressure[1000];
```

- Essentially defines 1000 variables ("array elements").
- Variables are named `pressure[0]`, `pressure[1]`, `pressure[2]`, …, `pressure[999]`
- General form:

```
data-type array-name[size];
```

- size also called length.
- `array-name[index]` gives `index`th variable.
- Necessary: `0 <= index < size`.
- **Not just a collection of variables:** Index may be given as an expression.

Array element operations

```
double pressure[1000];
cin >> pressure[0] >> pressure[2];
pressure[1] = (pressure[0] + pressure[2])/2;
for(int i=0; i<1000; i++) cin >> pressure[i];
cout << pressure[439]*3.33 << endl;
```

```
for(int i=1; i<999; i++)
  pressure[i] = (pressure[i-1] + pressure[i+1])/
2;
```

- Array index can be an expression which will be evaluated during execution and then the corresponding element will be used.
- Not possible with 1000 names.

Index out of range

```
double pressure[1000];
pressure[1000] = 1.2;
double d = pressure[-5];
```

- In the assignments above, the array index is outside the allowed range: 0 through size-1.
- If the array indes is out of range, the program may run and produce wrong results, may halt with a message. Nothing is guaranteed.

- The programmer must ensure index stays in range.

Initialization while defining

```
int squares[4] = {0, 1, 4, 9};

int cubes[] = {0, 1, 8, 27, 64};
// size = 5 inferred.


int x, pqr[200], y[]={1,2,3,4};
```