MA122 - Computer Programming and Apllications

Indian Institute of Space Science and Technology

January 27, 2017

Lecture 7

Compound Assignment

Expression	Equivalent to
$y+=x$	$y=y+x$
$y-=5$	$y=y-5$
$x/=y$	$x=x/y$
$z*=y+1$	$z=z*(y+1)$

Lecture 7

Operator Precedence

Precedence	Operator	Description	Associativity
2	++	Postfix increment	L-R
	−−	Postfix decrement	
3	++	Prefix increment	R-L
	−−	Prefix decrement	
5	*	Multiply	L-R
	/	Divide	
	%	Modulo	
6	+	Addition	L-R
	-	Subtraction	

Operator Precedence

Precedence	Operator	Description	Associativity
8	$<$	Less than	L-R
	$<=$	Less than or equal to	
	$>=$	greater than equal to	
	$>$	Greater than	
9	$==$	Equal to	L-R
	$!=$	Not equal to	
13	&&	Logical AND	L-R
14	||	Logical OR	L-R

Operator Precedence

Precedence	Operator	Description	Associativity
15	:?	Conditional	R-L
16	$=$	Simple assignment	R-L
	$*=$	Multiply and assign	
	$/=$	Divide and assign	
	$\%=$	modulo and assign	
	$+=$	Add and asign	
	$-=$	Subtract and assign	

Lecture 7

Introducing arrays

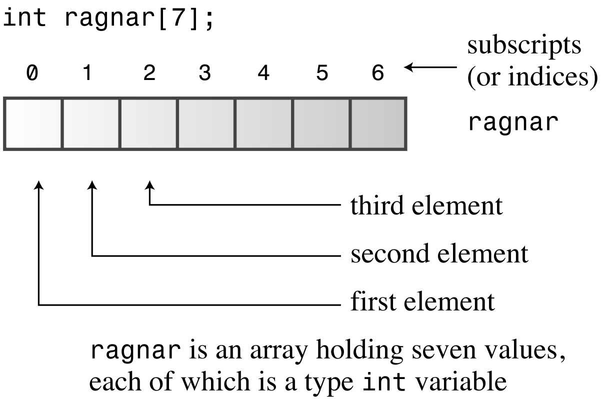

Figure 4.1 Creating an array.

Lecture 7

Introducing arrays continued...

```
// arrayone.cpp -- small arrays of integers
#include <iostream>
int main()
{
  using namespace std;
  int yams[3];   // creates array with three elements

  yams[0] = 7;   // assign value to first element
  yams[1] = 8;
  yams[2] = 6;

  int yamcosts[3] = {20, 30, 5}; // create, initialize
      array

  cout << "Total yams = ";
  cout << yams[0] + yams[1] + yams[2] << endl;
```

Introducing arrays continued...

```
cout << "The package with " << yams[1] << " yams
    costs ";
cout << yamcosts[1] << " cents per yam.\n";

int total = yams[0] * yamcosts[0] + yams[1] *
    yamcosts[1];
total = total + yams[2] * yamcosts[2];

cout << "The total yam expense is " << total << "
    cents.\n";

cout << "\nSize of yams array = " << sizeof yams;
cout << " bytes.\n";

cout << "Size of one element = " << sizeof yams[0];
cout << " bytes.\n";
return 0; }
```

array initialization

```
#include <iostream>
int main()
{
  using namespace std;

  int cards[4] = {3, 6, 8, 10};    // okay
  int hand[4];                     // okay

  hand[4] = {5, 6, 7, 9};          // not allowed
  hand = card;          // not allowed

  float hotelTips[5] = {5.0, 2.5}; //fewer values,
      allowed
```

array initialization continued ...

```

double earnings[4] {1.2e4, 1.6e4, 1.1e4, 1.7e4}; //
    okay with C++11

unsigned int counts[10] = {}; // all elements set to 0
float balances[100] {};      // all elements set to 0

long plifs[] = {25, 92, 3.0};        // not allowed

char slifs[4]= {'h', 'i', 1122011, '\0'}; // not
    allowed

char tlifs[4] ={'h', 'i', 112, '\0'}; // allowed

return 0; }
```