**CS 31**

**April 29th, 2020**

**Week 5**

**Lecture #10 : Arrays**

int

char

char ch = 76; // ‘L’ if ASCII is encoding, ‘V’ if EBCDIC

int k = ‘L’; // 76 if ASCII; 211 if EBCDIC

‘0’ is one less than ‘1’ which is one less than ‘2’...

‘0’ through ‘9’ are consecutive

‘a’ is less than ‘b’ which is less than ‘c’... ‘z’

‘A’ is less than ‘B’ which is less than ‘C’... ‘Z

ch++; // Now 77; ‘M’ if ASCII; ‘<’ if EBCDIC

k=77;

double d = 3.68;

cout.setf(ios:fixed);

cout.precision(1);

cout << d; // call the function for double : writes ‘3’ ‘.’ ‘7’

//if ASCII, this 51 46 55

//if EBCDIC, this 243 75 247

cout << k; //calls the function for int : writes ‘7’ ‘7’

//if ASCII, this is 55 55

//if EBCDIC, this is 247 247

cout << ch; //calls the function for char : writes ‘M’ if ASCII, ‘C’ if EBCDIC

//if ASCII, this is 77

//if EBCDIC, this is 77

string s = “A7”;

cout << s; //calls the function for string : writes ‘A’ ‘7’

//if ASCII, this is 65 55

//if EBCDIC, this is 193 247

|  |
| --- |
| int main(){  int day;  cin >> day;  if(isValidDate(2020,4,day))  ... } |

|  |
| --- |
| bool isValidDate(int y, int m, int d){  ...  if(m < 1 || m > 12 || d < 1)  return false;  if (m==1 || m==3 || m==5 || m ==7 || m == 8 || m ==10 || m==12)  return d <=31;  if(m==4 || m==6 || m==9 || m==11)  return d <=30;  //At this point m must be 2  ... deal with the leap year issue and return...  } |

|  |
| --- |
| bool isValidDate(int y, int m, int d){  int daysInMonth[12] = { 31, 28, 31, 30, 31, 30,  31, 31, 30, 31, 30, 31 };  ...   if(m<1 || m>12 || d<1)  return false;  if(m!=2)  return d<= daysInMonth [m-1];  ///at this point, m must be 2  ...deal with leap/nonleap year and return... } |

|  |
| --- |
| switch (m){  case 1: cout << "January"; break;  case 2: cout << "February"; break;  ...  case 12: cout << "December"; break; } |

|  |
| --- |
| const int NMONTHS=12; const string monthNames[NMONTHS] = {  "January", "February", ...., "December" };  //cout << monthNames[m-1];  const int daysInMonth[NMONTHS] = { 31, 28, 31, 30, 31, 30,  31, 31, 30, 31, 30, 31 };  cout << "These months have 31 days : "; for (int k = 0; k < NMONTHS; k++){  if (daysInMonth[k] == 31)  cout << monthNames[k] <<endl; } |

[0] [1] [2] ...

31 28 31 30 31 30 31 31 30 31 30 12

if(...)

… true part …;

else

… false part…;

… next statement …

… daysInMonth[-1]

… daysInMonth[12]

… daysInMonth[13]

Undefined behavior

Biggest mistake: not guaranteeing that code will access elements of the array within bounds

k < daysInMonth.size()

k < daysInMonth.length()

CANNOT DO THIS WITH ARRAYS IN C++, cannot ask how big an array is in most cases

daysInMonth.at(k) → cannot do this with arrays

Array size: number of elements must be a constant at the time of compiling program

|  |
| --- |
| cout << "How many scores will you enter? "; **int** nScores; cin >> nScores;  cout << "Enter the scores: " << endl; |

Bad design: cannot guarantee that user can count exactly and input all scores

|  |
| --- |
| int nScores = 0; cout << "Enter the scores (negative when done):" << endl;  for ( ; ; ) {  int s;  cin >> s;  if (s < 0)  break;  total += s;  nScores++; } |