**CS 31**

**May 4th, 2020**

**Week 6**

**Lecture #11 : More Arrays**

|  |
| --- |
| 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

cout << “Give me an upper limit for the number of scores? “;

int limit;

int scores[limit]; //Error in standard C++ -- number of elements must be known at compile time

|  |
| --- |
| #include<cmath> ...  const int MAX\_NUMBER\_OF\_SCORES = 10000; int scores[MAX\_NUMBER\_OF\_SCORES]; int nScores = 0; int total=0; cin >> limit; cout << "Enter the scores (negative when done):" << endl; for (;;){  int s;  cin >> s;  if (s < 0)  break;  if(nscores==MAX\_NUMBER\_OF\_ITEMS){  cout << "I can handle only " << MAX\_NUMBER\_OF\_SCORES << " scores!" << endl;  cout << "Continuing with only the first " <<MAX\_NUMBER\_OF\_SCORES << " values." << endl;  break;  } scores[nScores] = s;  total += s;  nScores++; } if(nScores ==0)  cout << "There were no scores, so no statistics" << endl; else{  cout << "the mean of the scores is " << static\_cast<double>(total)/nScores << endl;  double sumOfSquares = 0;  for(int k=0; k<nScores;k++){  double diff = scores[k]- mean;  sumOfSquares += diff\* diff;  }  cout << "The std. deviation is " << sqrt(sumOfSquares/nScores) << endl; } |

0 1 2 3 4 5 9998 9999

scores : 90 87 94 76 95 83 … 84 78

nscores : 10000

total : ...

|  |
| --- |
| double computeMean(const int a[], int n); void setAll(int a[], int n, int value);  int main(){  const int MAX\_NUMBER\_OF\_SCORES = 10000;  int scores[MAX\_NUMBER\_OF\_SCORES];  int nScores =0;  ... fill up the array (perhaps partially) ..  double m = computeMean(scores, nScores);  ...  int stuff[100];  ...fill up all 100 elements of stuff  double m2 = computeMean(stuff, 100);  ...  const int daysInMonth[12] = {31, 28, ..., 31};  cout << computeMean(daysInMonth, 12);  set(stuff, 50, -42);   set(daysInMonth, 12, 30); //Error! Won't compile! }  double computeMean(int a[],const int n){  if(n<=0)  return 0;  int total = 0;  for(int k=0; k<n; k++)  total += a[k];  return static\_cast<double>(total) / n; }  void setAll(int a[], int n, int value){  for(int k=0; k<n; k++){  a[k]=value;  } } |