

Practice for Exam 2

Randomness

- Write code to seed the random number generator with the current time
- Write code to seed the random number generator with user input
- Write code to seed the random number generator with the number 12.
- Write code to get a random number from 1 to 20
- Write code to get a random number from 200-300

Using Functions (review via Exam 1):

- Write a program to find $f(x)$ given an x by the user.

$f(x)$ is $0.5 * \sin(x)^2 + 0.5 * \cos(x)^2$

What is interesting about the result?

- Write a program that outputs a random integer from 13 to 23.
- Write a program to output the natural logarithm of x . Get the input of x from the user. Ensure that x is greater than 0.

Writing User-Defined Functions:

- Write a function to compute the maximum of 3 integers.
- Write a function to compute the result of $f(x)$ given above.
- Write a function that computes the distance moved by an object with a given velocity and acceleration over a period of time. The equation to compute this is:

$\text{distance} = \text{velocity} * \text{time} + 0.5 * \text{acceleration} * \text{time}^2$

- Use call by reference to write an input function to read in an integer. Use this function prototype:
`void getInteger(string, int&)`

Vectors:

- Create a vector of strings called names with 5 elements. Create another vector of integer grades with 5 elements. Read in the name and grades from the user. Find the average grade and standard deviation of the grades. Standard deviation is computed by first adding up each elements difference with the average and squaring it. For example, if the average is avg, then compute:
 $(\text{grades}[i] - \text{avg})^2$
for each element and sum it all up. We'll call this sum, just sum. Then, given the sum, compute:
 $\text{stddev} = \sqrt{0.2 * \text{sum}}$
The 0.2 here is computed by 1 over the number of elements. In this case, $1/5 = 0.2$
When complete, output the average, and standard deviation. Then, for each element, report how many standard deviations away from the average each grade is by computing: $(\text{grades}[i] - \text{avg}) / \text{stddev}$
- Take a vector of 30 integers and reverse their order, such that the last becomes the first.

- Take a vector of 30 integers and move every element down one position (so #2 goes into #1). The first element should then be placed in the last slot.
- Allow the user to compute the average and standard deviation of an arbitrary number of integer grades. Let the user choose how many and let the user enter the values.
- Create a vector of strings called birds with as many elements as the user wishes. Then, randomly choose two elements from this vector and output their sum. (Sums on strings performs a concatenation.) Let the user choose how many birds and enter the bird strings.