

CS 111

calling functions, sqrt(), rand(), srand()

Library functions

- C++ comes with lots of predefined functions that we can use when we need them
 - These functions are stored in libraries
- For now, we concentrate on the `cmath` library that has many useful math functions, including `sqrt()`

Examples

- `double squareRoot = sqrt(5.0);`
 - function argument is 5.0
 - function return type is double
- `srand(time(0));`
 - function takes an int as an argument
 - the time function returns the number of seconds since 00:00 hours, Jan 1, 1970 UTC
 - `srand` itself returns no value (i.e., it is return type void)
- `int score = rand();`
 - function does not take any arguments
 - function return type is int

Using sqrt(), rand(), and srand()

- To use sqrt():
 - add `#include<cmath>` below `#include<iostream>`
- To use rand():
 - add `#include<ctime>` and `#include<cstdlib>` below `#include<iostream>`
 - before calling `rand()`, add this line to your code before the function call:
`srand(time(0));`
- `srand()` is used in conjunction with the `rand()` function to generate a distinct sequence of random numbers each time `rand()` is called

Generate random number in range

- To generate a random number within a range from some value min to some value max, use this formula
 - $\text{min} + \text{rand}() \% (\text{max} - \text{min} + 1)$
- Example: Generate a random number between 35 and 90
 - $35 + \text{rand}() \% (90 - 35 + 1) = 35 + \text{rand}() \% (56)$

13.3 Pseudocode

// List all numbers that are perfect squares between two randomly generated numbers, each within a specific range

Generate random number in low range and store in int variable low

Generate random number in high range and store in int variable high

Set up loop from low to high

 Set int root equal to square root of counter

 If $\text{root} * \text{root}$ equals the counter value, print counter value