Students:
Section 2.19 is a part of 2 assignments: CSC108 CH02.11-2.24 C2B

This assignment's due date has passed. Activity will still be recorded, but will not count towards this assignment (unless the due date is changed). See this article for more info.

Includes: CA
Due: 02/06/2025, 11:59 PM EST

2.19 Random numbers

Generating a random number

Credit: Microsoft ClipArt gallery

Some programs need to use a random number. Ex: A game program may need to roll dice, or a website program may generate a random initial password.

The ***rand()*** function, in the C standard library, returns a random integer each time the function is called, in the range 0 to RAND_MAX.

Figure 2.19.1: Outputting three random integers.

```
#include <iostream>
#include <cstdlib>
using namespace std;

int main() {
   cout << rand() << endl;
   cout << rand() << endl;
   cout << rand() << endl;

   cout << "(RAND_MAX: " << RAND_MAX << ")" << endl;

   return 0;
}
```

```
16807
282475249
1622650073
(RAND_MAX: 2147483647)
```

Feedback?

Line 2 includes the C standard library, which defines the rand() function and RAND_MAX.

RAND_MAX is a machine-dependent value, but is at least 32,767. Above, RAND_MAX is about 2 billion.

Usually, a programmer wants a random integer restricted to a specific number of possible values. The modulo operator % can be used. Ex: `integer % 10` has 10 possible remainders: 0, 1, 2, ..., 8, 9.

PARTICIPATION ACTIVITY 2.19.1: Restricting random integers to a specific number of possible values.

Start 2x speed

rand() % 3   Possible remainders are 0, 1, 2

24 % 3   0
22457 % 3   2

0 % 3 = 0
1 % 3 = 1
2 % 3 = 2
3 % 3 = 0
4 % 3 = 1
5 % 3 = 2
6 % 3 = 0
...

rand() % N yields N possible values, from 0 to N-1

Captions

1. Each call to rand() returns a random integer between 0 and a large number RAND_MAX.
2. A programmer usually wants a smaller number of possible values, for which % can be used. % (modulo) means remainder. rand() % 3 has possible remainders of 0, 1, and 2.
3. Thus, rand() % 3 yields 3 possible values: 0, 1, and 2. Generally, rand() % N yields N possible values, from 0 to N-1.

Feedback?

PARTICIPATION ACTIVITY 2.19.2: Random number basics.

1) What library must be included to use the rand() function?
   - The C random numbers library
   - The C standard library

2) The random integer returned by rand() will be in what range?
   - 0 to 9
   - -RAND_MAX to RAND_MAX
   - 0 to RAND_MAX

3) Which expression's range is restricted from 0 to 7?
   - rand() % 7
   - rand() % 8

4) Which expression yields one of 5 possible values?
   - rand() % 4
   - rand() % 5
   - rand() % 6

5) Which expression yields one of 100 possible values?
   - rand() % 99
   - rand() % 100
   - rand() % 101

6) Which expression would best mimic the random outcome of flipping a coin?
   - rand() % 1
   - rand() % 2
   - rand() % 3

7) What is the smallest *possible* value returned by rand() % 10?
   - 0
   - 1
   - 10
   - Unknown

8) What is the largest *possible* value returned by rand() % 10?
   - 10
   - 9
   - 11

Feedback?

Specific ranges

The technique above generates random integers with N possible values ranging from 0 to N-1, like 6 values from 0 to 5. Commonly, a programmer wants a specific range that starts with some value x that isn't 0, like 10 to 15, or -20 to 20. The programmer should first determine the number of values in the range, generate a random integer with that number of possible values, and then add x to adjust the range to start with x.

PARTICIPATION ACTIVITY 2.19.3: Generating random integers in a specific range not starting from 0.

Start 2x speed

10 11 12 13 14 15
15 - 10 + 1   6 possible values

rand() % 6
0 1 2 3 4 5

(rand() % 6) + 10
10 11 12 13 14 15

Captions

1. A programmer wants random integers in the range 10 to 15. The number of possible values is 15 − 10 + 1. (People often forget the + 1.)
2. rand() % 6 generates 6 possible values as desired, but with range 0 to 5.
3. Adding 10 still generates 6 values, but now those values start at 10. The range thus becomes 10 to 15.

Feedback?

PARTICIPATION ACTIVITY 2.19.4: Generating random integers in a specific range.

1) Goal: Random integer from the 6 possible values 0 to 5.
   rand() % ___

   Check   Show answer

2) Goal: Random integer from 0 to 4.
   rand() % ___

   Check   Show answer

3) How many values exist in the range 10 to 15?

   Check   Show answer

4) How many values exist in the range 10 to 100?

   Check   Show answer

5) Goal: Random integer in the range 10 to 15.
   (rand() % 6) + ___

   Check   Show answer

6) Goal: Random integer in the range 16 to 25.
   (rand() % ___) + 16

   Check   Show answer

7) How many values are in the range -5 to 5?

   Check   Show answer

8) Goal: Random integer in the range -20 to 20.
   (rand() % 41) + ___

   Check   Show answer

Feedback?

PARTICIPATION ACTIVITY 2.19.5: Specific range.

1) Which generates a random integer in the range 18 ... 30?
   - rand() % 30
   - rand() % 31
   - rand() % (30 - 18)
   - (rand() % (30 - 18)) + 18
   - (rand() % (30 - 18 + 1)) + 18

Feedback?

The following program randomly moves a student from one seat to another seat in a lecture hall, perhaps to randomly move students before an exam. The seats are in 20 rows numbered 1 to 20. Each row has 30 seats (columns) numbered 1 to 30. The student should be moved from the left side (columns 1 to 15) to the right side (columns 16 to 30).

Figure 2.19.2: Randomly moving a student from one seat to another.

```
#include <iostream>
#include <cstdlib>
using namespace std;

// Switch a student
// from a random seat on the left  (cols  1 to 15)
//   to a random seat on the right (cols 16 to 30)
// Seat rows are 1 to 20

int main() {
   int rowNumL;
   int colNumL;
   int rowNumR;
   int colNumR;

   rowNumL = (rand() % 20) + 1;  // 1 to 20
   colNumL = (rand() % 15) + 1;  // 1 to 15

   rowNumR = (rand() % 20) + 1;  // 1 to 20
   colNumR = (rand() % 15) + 16; // 16 to 30

   cout << "Move from ";
   cout << "row " << rowNumL << " col " << colNumL;
   cout << " to " ;
   cout << "row " << rowNumR << " col " << colNumR;
   cout << endl;

   return 0;
}
```

```
Move from row 8 col 5 to row 14 col 24
```

Feedback?

PARTICIPATION ACTIVITY 2.19.6: Random integer example: Moving seats.

Consider the above example.

1) The row is chosen using (rand() % 20) + 1. The 20 is because 20 rows exist. The + 1 is _____ .
   - necessary
   - optional

2) The column for the left is chosen using (rand() % 15) + 1. The 15 is used because the left half of the hall has _____ columns.
   - 15
   - 30

3) The column for the right could have been chosen using (rand() % 15) + 15.
   - True
   - False

Feedback?

Pseudo-random

The integers generated by rand() are known as pseudo-random. "Pseudo" means "not actually, but having the appearance of". The integers are pseudo-random because each time a program runs, calls to rand() yield the same sequence of values. Earlier in this section, a program called rand() three times and output 16807, 282475249, 1622650073. Every time the program is run, those same three integers will be printed. Such reproducibility is important for testing some programs. (Players of classic arcade games like Pac-man may notice that the seemingly-random actions of objects actually follow the same pattern every time the game is played, allowing players to master the game by repeating the same winning actions).

Internally, the rand() function has an equation to compute the next "random" integer from the previous one, (invisibly) keeping track of the previous one. For the first call to rand(), no previous random integer exists, so the function uses a built-in integer known as the ***seed***. By default, the seed is 1. A programmer can change the seed using the function srand(), as in srand(2) or srand(99).

If the seed is different for each program run, the program will get a unique sequence. One way to get a different seed for each program run is to use the current time as the seed. The function ***time()*** returns the number of seconds since Jan 1, 1970.

Note that the seeding should only be done once in a program, before the first call to rand().

Figure 2.19.3: Using a unique seed for each program run.

```
#include <iostream>
#include <cstdlib>
#include <ctime>    // Enables use of time() function
using namespace std;

int main() {
   srand(time(0));  // Unique seed
   cout << rand() << endl;
   cout << rand() << endl;
   cout << rand() << endl;

   return 0;
}
```

```
636952311
51510682
304122633

(next run)

637053153
1746362176
1450088483
```

Feedback?

PARTICIPATION ACTIVITY 2.19.7: Using a unique seed for each program run.

1) The s in srand() most likely stands for _____ .
   - sequence
   - seed

2) By starting a program with srand(15), calls to rand() will yield a different integer sequence for each program run.
   - True
   - False

3) By starting a program with srand(time(0)), calls to rand() will yield a different integer sequence for each successive program run.
   - True
   - False

4) rand() is known as generating a "pseudo-random" sequence of values because the sequence begins repeating itself after about 20 numbers.
   - True
   - False

Feedback?

<random> number library

The <random> number library provides greater control of the random-number generation. Using the library, a programmer may specify the engine used to produce the numbers and the distribution of the numbers. Trivial random-number sequences may still be generated using <cstdlib> library functions srand() and rand().

Exploring further:

- C++ <random> number library

CHALLENGE ACTIVITY 2.19.1: Generate a random integer.

620890.5010016.qx3zqy7

Start

Generate a random integer between 0 and 3 (inclusive)

rand() % Ex: 20

1 2 3 4

Check   Next

Feedback?

CHALLENGE ACTIVITY 2.19.2: Random numbers.

620890.5010016.qx3zqy7

Start

Integer seedVal is read from input. srand() is called with seedVal as the seed. Use rand() to assign variables data1, data2, data3, and data4 each with a random number generated between 0 and 11, both inclusive.

Click here for example

```
#include <iostream>
#include <cstdlib>
using namespace std;

int main() {
   int seedVal;
   int data1;
   int data2;
   int data3;
   int data4;
   int sum;

   cin >> seedVal;

   srand(seedVal);

   /* Your code goes here */
```

1 2 3

Check   Next level

Feedback?

How was this section?   Provide section feedback

Activity summary for assignment: CSC108 CH02.11-2.24 C2B   51 / 51 points

Due: 02/06/2025, 11:59 PM EST   51 / 51 points submitted to BlackboardLearn

This assignment's due date has passed. Activity will still be recorded, but will not count towards this assignment (unless the due date is changed). See this article for more info.

Completion details