# Laboratory 03A

CS-102

W. M. Littlefield

# Laboratory 3A
# Part 1

- Your assignment is to write a program that plays a word game with the user.  The program should ask the user to enter the following:
    - Your full name (first name and last name with a space in between)
    - Your age (in numbers) when you started attending college
    - The name of the city you live in
    - The name of the college you go to
    - Your profession
    - The type of animal that you like
    - Your pet’s name (or what you would name it if you had one)

# Laboratory 3A

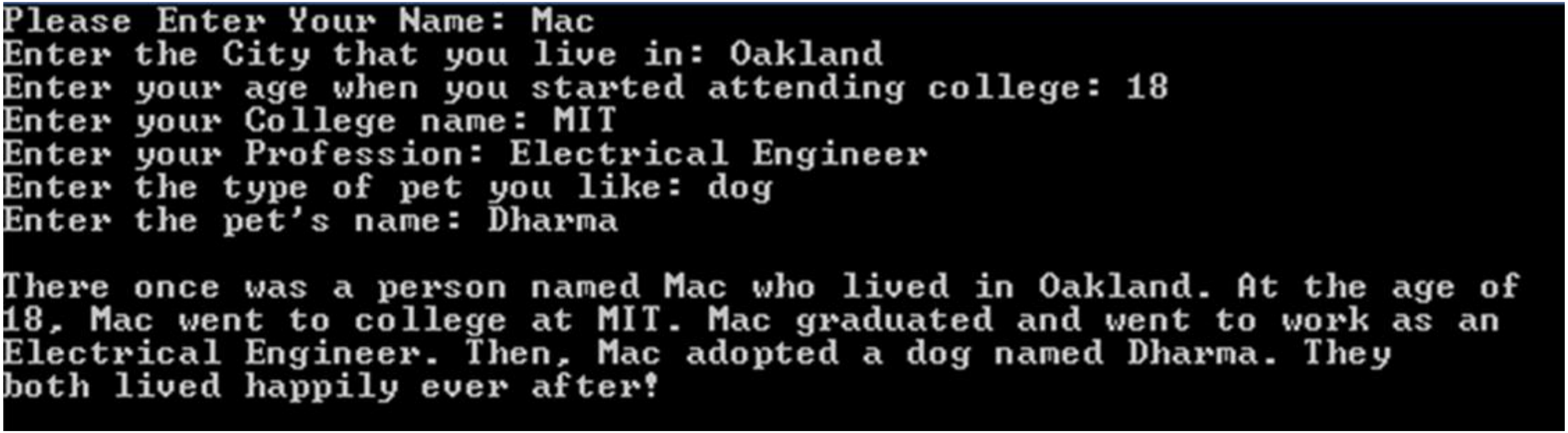
- After the user has entered these items, the program should display the following story, inserting the user's input into the appropriate locations.

- Type your variables as follows:
  - **string name, city, college, profession, animal, petName;**
  - **int age;**

- Your resulting story:
  - There once was a person named ***name*** who lived in ***city*** . At the age of ***age*** , ***name*** went to college at ***college.*** ***name*** graduated and went to work as a(n) ***profession.*** Then, ***name*** adopted a(n) ***animal*** named ***petName.*** They both lived happily ever after!

# Laboratory 3A

- Here is a sample input to and output from the program:



```
Please Enter Your Name: Mac
Enter the City that you live in: Oakland
Enter your age when you started attending college: 18
Enter your College name: MIT
Enter your Profession: Electrical Engineer
Enter the type of pet you like: dog
Enter the pet's name: Dharma

There once was a person named Mac who lived in Oakland. At the age of
18, Mac went to college at MIT. Mac graduated and went to work as an
Electrical Engineer. Then, Mac adopted a dog named Dharma. They
both lived happily ever after!
```

- Call your program: *YourName*_Lab03A-1.cpp

# Laboratory 3A

- Take Note:
  - Be very careful of the spacing.  Don’t run words together or have more than one space between words.
  - **Don’t use more than 80 characters per line** (i.e. don’t let your lines wrap to another line because they are too long).
  - If your profession starts with a vowel, precede it with **an**, otherwise use an **a**.
  - Likewise, if your animal type begins with a vowel, use **an**, otherwise use an **a**.
  - Don’t forget to put your name on your program.
  - When you are ready:
    - If you are doing the Lab synchronously, call the Instructor over so your work can be checked and you can be given credit.
    - If you are doing the lab asynchronously, submit *YourName*_Lab03A-1.cpp to Canvas.

# Laboratory 3A
# Part 2

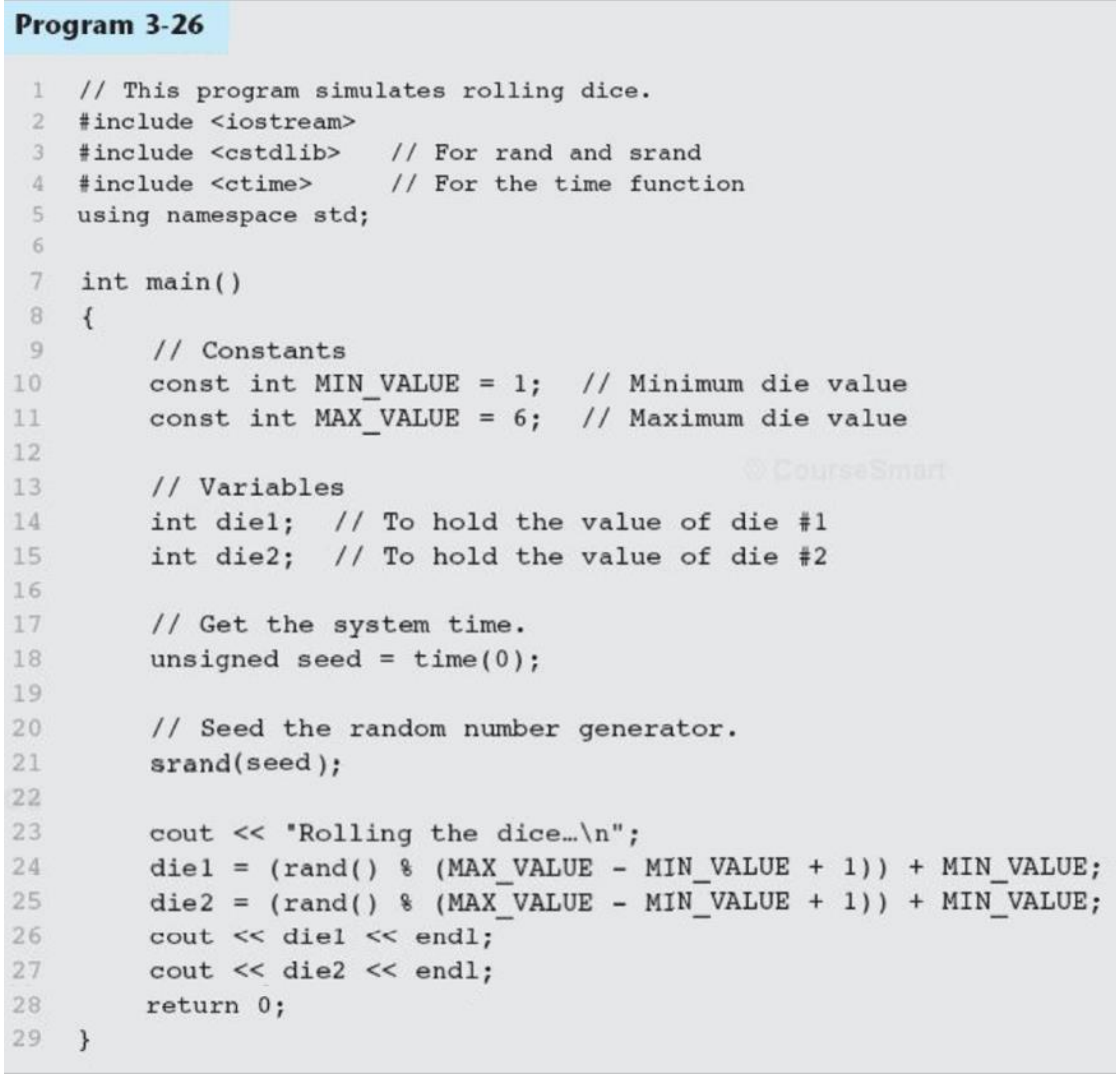
- In the lecture you saw how to use the pseudo-random number generator to generate a dice throwing program.
- You are being asked to write a program that will flip a Coin.
- Let a 0 be considered Tails, and a 1 be considered Heads.
- Using Program 3-26 as a model, write a program that will flip a Coin and display the result.
- Call your program: *YourName*_Lab03A-2.cpp

# Laboratory 3A
# Part 2

- Assuming you called your Coin, “coin”, you may add the following code to yield a text output:

```
if (coin == 1)
	cout << "HEADS" << endl;
else
	cout << "TAILS" << endl;
```

- When you have your coin flipping program working:
  - If you are doing this Lab synchronously, call over the instructor so that you may receive proper credit.
  - If you are doing this Lab asynchronously, submit your program to Canvas.



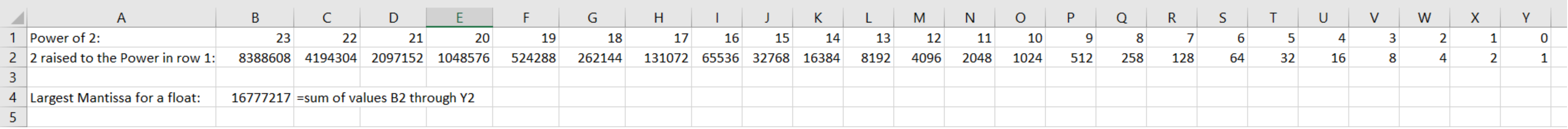
**Program 3-26**

```
// This program simulates rolling dice.
#include <iostream>
#include <cstdlib>   // For rand and srand
#include <ctime>     // For the time function
using namespace std;

int main()
{
    // Constants
    const int MIN_VALUE = 1;  // Minimum die value
    const int MAX_VALUE = 6;  // Maximum die value

    // Variables
    int die1;  // To hold the value of die #1
    int die2;  // To hold the value of die #2

    // Get the system time.
    unsigned seed = time(0);

    // Seed the random number generator.
    srand(seed);

    cout << "Rolling the dice…\n";
    die1 = (rand() % (MAX_VALUE - MIN_VALUE + 1)) + MIN_VALUE;
    die2 = (rand() % (MAX_VALUE - MIN_VALUE + 1)) + MIN_VALUE;
    cout << die1 << endl;
    cout << die2 << endl;
    return 0;
}
```

# Laboratory 3A
# Part 3

- The purpose of this exercise is to acquaint you with the number of significant digits you have available to you in the Mantissa of a floating point number.
- For example, the float type has a mantissa of 24 bits, one of which is a sign bit. This means that you have 23 binary bits of accuracy for your float type numbers.
- 23 bits of accuracy means that the largest number that can be in a float mantissa is: 16,777,217 (see following page)
- By setting all 23 bits to 1, and then adding up the result, we get this largest value.
- For numbers larger than this, the mantissa starts dropping the least significant bits.

# Largest Float Mantissa

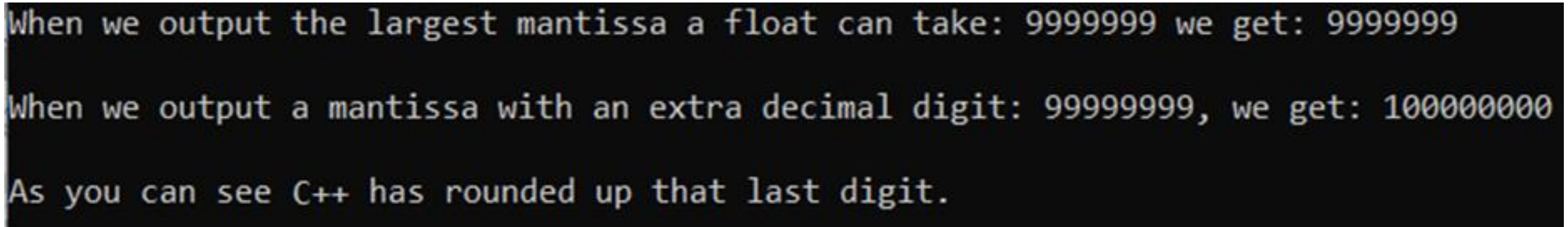


| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Power of 2: | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 2 | 2 raised to the Power in row 1: | 8388608 | 4194304 | 2097152 | 1048576 | 524288 | 262144 | 131072 | 65536 | 32768 | 16384 | 8192 | 4096 | 2048 | 1024 | 512 | 258 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 3 | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | Largest Mantissa for a float: | 16777217 | =sum of values B2 through Y2 | | | | | | | | | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | | | | | | | | | | | |

As you can see: 16,777,217 contains 8 decimal digits. However, because no number can be greater than 16,777,217 we would conservatively say that the largest number that a float can handle with exact precision is 7 decimal digits in size.

# If you run the following program you will see that that when we exceed 7 digits of mantissa we get round-off errors

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
   float longest = 9999999.0;
   float toolong = 99999999.0;
   cout << "When we output the largest mantissa a float can take: 9999999 we get: ";
   cout << setprecision(9) << longest << endl << endl;
   cout << "When we output a mantissa with an extra decimal digit: 99999999, we get: ";
   cout << setprecision(9) << toolong << endl << endl;
   cout << "As you can see the C++ compiler has rounded up that last digit." << endl;
   cin.get();
   return 0;
}
```



```
When we output the largest mantissa a float can take: 9999999 we get: 9999999

When we output a mantissa with an extra decimal digit: 99999999, we get: 100000000

As you can see C++ has rounded up that last digit.
```

# Laboratory 3A
# Part 3

- Now do this same exercise for a double floating point type.
- What are the number of significant bits that the double’s mantissa can hold, excluding the sign bit.
- What is the largest number that a double’s mantissa can hold without roundoff error?
- Repeat the program shown on the previous page but set to show where the double’s mantissa starts to exhibit roundoff errors.
- Call the program: *YourName*_Lab03A-3.cpp
- When you have your result:
  - If you are doing this Lab synchronously, call over the instructor so that you may receive proper credit.
  - If you are doing this Lab asynchronously, submit your program to Canvas.