

## Section 5: Random Number Selection

**Directions:** Copy the following program. Complete the questions and challenge.

```
1  #include<iostream>
2  #include <stdlib.h>    //srand, rand
3  #include <time.h>     //allows time to generate a random number at a certain point
4  using namespace std;
5
6  int main()
7  {
8      char a;
9      int iSecret, iGuess;
10
11     srand (time(NULL));    // initialize random seed so a different number can be picked each time
12
13     iSecret = rand() % 10 + 1;    //generate secret number between 1 and 10:
14
15     do {
16         cout << "Guess the number (1 to 10): ";
17         cin >> iGuess;
18
19         if (iSecret<iGuess)
20         {
21             cout << "The secret number is lower";
22         }
23         else if (iSecret>iGuess)
24         {
25             cout << "The secret number is higher";
26         }
27     } while (iSecret!=iGuess);
28
29     cout << "Congratulations!";
30     return 0;
31 }
```

### Explanation on creating a random number

Adding the <stdlib.h> library allows the programmer to generate a random number

srand (time(NULL)); allows the program to generate a new random number each time through the program. The random number generator is based on the current time the program is compiled.

- time sets the random number generator in the <stdlib.h> library sets the current time and erasing the previous random generated number that ran at a previous time.
- NULL represents a pointer. C++ needs a pointer at a specific spot to point to. Since we are not worried about pointing at a specific time a NULL value will work.

### Questions/Challenge

1. Challenge: Modify the program so the user only gets 3 guesses.
  - a. Output the following
    - i. If the user guesses the right answer on the first try  
Output: You Gussed right 1<sup>st</sup> time!
    - ii. If the user guesses the right answer on the second try  
Output: Good Job!
    - iii. If the user guesses the right answer on the third try  
Output: Not bad it took you 3 guesses
    - iv. If the user does not guess the right answer  
Output: Better luck next time.

### Program 2: Combination Lock Program

1. Write a program that has them enter a 3 digit combination ranging from numbers 1-5
2. Have the program Randomly Generate a combination of 3 numbers
3. User may only try 3 times to guess the correct combination
4. Each time a the user inputs a number have the program tell them if the number inputted is the correct number for that place in the combination
5. Final Output
  - a. If the combination is inputted correctly  
Output: Unlocked
  - b. If the combination is not inputted correctly  
Output: You are locked out. Contact an Administrator

Hint on Check Program: May display random generated combination to check if it works

