Functions, Pass by Reference and Default Parameters

This homework will demonstrate the use of pass by reference and default parameters with functions. We will use default parameters to let a print vector function print in reverse.

Requirements

- You will be using the following 2 functions to load a vector with random values in the range of 100 and print that vector using pass by reference:

```cpp
// Load vector with indicated number of values from 0 to range-1
vector<int> loadVector(int numValues, int range) {
    vector<int> values;
    for (int i = 0; i < numValues; i++) {
        int val = rand() % range;
        values.push_back(val);
    }
    sort(values.begin(), values.end());
    return values;
}

// print vector with indicated number of values from 0 to range-1
void printVector(const vector<int>& nums) {
    for (int i = 0; i < nums.size(); i++) {
        cout << nums.at(i) << endl;
    }
    cout << endl << endl;
}
```

- Include following header files `<vector>`, `<ctime>` and `<algorithm`
- Create 2 `constant global int variables` called NUM_VALUES =10, and RANGE = 100
- Create a random seed based on time in the main() method for the random number generator using function srand(time(NULL));
- Create necessary function calls in main() for loadVector(NUM_VALUES, RANGE) and printVector(nums); Note that vector<int> nums is returned form the loadVector method.
- Create another printVector function to take a default boolean parameter called reverse. Add necessary function call in main(). Set your default value for the function to false.
  - When reverse is false, the vector is printed from index 0 to index size()-1
  - When reverse is true, the vector is printed from index size()-1 to 0
  - Hint: When printing reverse nums.at(nums.size()-1 -i) can be used.
- Make sure you have your name and Bronco ID in the header comment
  /* Name: Jane-Joe
   * Bronco ID: 12345678
   * Sources of Help: Jon Doe helped me with............
   */

Due: November 23, 2016 by 4.00 PM. submit your source.cpp file to Blackboard.

Total Points = 100

- Correctness/Robustness: 60 points
- Code complies to requirements: 20 points
- Good coding style: 20 points