CMPE 285 – Software Engineering Processes

Lab 10 - Code Review

Group Members:
Harish Marepalli (016707314)
Dodda Sai Venkata Phanith (016725280)
Sourya Prateek Jammula (016702062)
Tirumala Saiteja Goruganthu (016707210)
Venkata Subramanya Sri Raviraj Indraganti (016254823)

Examples from various external sources:

Code Sample 1:

```
public static int dayOfYear(int month, int dayOfMonth, int year) {
   if (month == 2) {
      dayOfMonth += 31;
   } else if (month == 3) {
      dayOfMonth += 59;
   } else if (month == 4) {
      dayOfMonth += 90;
   } else if (month == 5) {
      dayOfMonth += 31 + 28 + 31 + 30;
   } else if (month == 6) {
      dayOfMonth += 31 + 28 + 31 + 30 + 31;
   } else if (month == 7) {
      dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30;
   } else if (month == 8) {
      dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31;
   } else if (month == 9) {
      dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31;
   } else if (month == 10) {
      dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30;
   } else if (month == 11) {
      dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30 + 31;
   } else if (month == 12) {
      dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30 + 31 + 31;
   }
   return dayOfMonth;
```

```
}
```

Changes Suggested:

1. Variable year should be used to check if it's a leap year - and accordingly update the number of days in February.
2. Code duplication can be avoided (example: number of days in March is written a lot of times) - have a dictionary storing the months and the number of days.
3. Avoid writing random numbers (Magic Numbers) like -
   ```
   else if (month == 4) {
           dayOfMonth += 90;
    }
   ```
   This affects the readability of the code.
4. Write comments in the code. It's not clear how the number "90" appears out of nowhere - affects the readability of the code.
5. The return value of the function, which is not the day of the month but should be day of the year.
6. Check the range of month, day, year. Different parts of the world have different formats for day/month/year. If we invoke this code in an incorrect way, it will still work. But the expected behavior is it should fail with an appropriate error message.

Code Sample 2:

```
public static int LONG_WORD_LENGTH = 5;
public static String longestWord;

public static void countLongWords(List words) {
  int n = 0;
  longestWord = "";
  for (String word: words) {
     if (word.length() > LONG_WORD_LENGTH) ++n;
     if (word.length() > longestWord.length()) longestWord = word;
  }
  System.out.println(n);
}
```

Changes Suggested:

1. Method doesn't return an integer despite having an integer return type. If the method is required to print the result, then the return type should be void. So here Method **countLongWords** should return variable **n**, not print it.
2. **LONG_WORD_LENGTH** is a global variable. Use of global variables is risky because a **public** modifier makes it accessible anywhere, and **static** means there is a single instance of the variable. Hence instead of defining a global variable, we can pass it in the function parameter, or we can change the modifier to **public static final** which makes **LONG_WORD_LENGTH** a global constant which makes the variable immutable and accessible anywhere in the code.

Code Sample 3:

```
# Python program for implementation of Bubble Sort

def bubbleSort(arr):
    n = len(arr)

    # Traverse through all array elements
    for i in range(n-1):
    # range(n) also work but outer loop will repeat one time more than needed.

        # Last i elements are already in place
        for j in range(0, n-i-1):

            # traverse the array from 0 to n-i-1
            # Swap if the element found is greater
            # than the next element
            if arr[j] > arr[j+1] :
                arr[j], arr[j+1] = arr[j+1], arr[j]

# Driver code to test above
arr = [64, 34, 25, 12, 22, 11, 90]

bubbleSort(arr)

print ("Sorted array is:")
for i in range(len(arr)):
    print ("%d" %arr[i]),
```

Changes Suggested:

1. If the array is already sorted then we don't need to traverse again, it will break the loop if it is already sorted. So, we can have a flag to keep track of the sorted part of the array so that we don't need to traverse that part again. As shown in the below code:

```
def bubbleSort(arr):
    n = len(arr)

    # Traverse through all array elements
    for i in range(n-1):
        swapped = False

        # Last i elements are already
        #  in place
        for j in range(0, n-i-1):

            # traverse the array from 0 to
            # n-i-1. Swap if the element
            # found is greater than the
            # next element
            if arr[j] > arr[j+1] :
                arr[j], arr[j+1] = arr[j+1], arr[j]
                swapped = True

        # IF no two elements were swapped
        # by inner loop, then break
        if swapped == False:
            Break
```