Assignment 2

Mobile Application Development (Comp 463)

(Java Practice)

Deadline: Monday Sept, 28st. In class (No late submissions will be accepted)

Total Points: 60

Important Notes: This is an individual assignment. Code sharing, reusing code on the internet or taking help from fellow students is strictly prohibited. Violators will receive no points not only for this assignment but also for the next 3 homework assignments.

Question 1 (30 Points)

A class called Author is designed as follows:

- Three private instance variables: name (String), email (String), and gender (char of either 'm' or 'f');
- One constructor to initialize the name, email and gender with the given values;

```
public Author (String name, String email, char gender) {......}
```

(There is no default constructor for Author, as there are no defaults for name, email and gender.)

- public getters/setters: getName(), getEmail(), setEmail(), and getGender(); (There are no setters for name and gender, as these attributes cannot be changed.)
- A toString() method that returns a string in this format : "*author-name (gender) at email*", e.g., "Sameera Ghayyur (f) at sameeraghayyur@fccollege.edu.pk".

Write the Author class.

Also write a *test program* called TestAuthor to test the constructor and all public methods. Try changing the email of an author, e.g.,

```
Author anAuthor = new Author("Tan Ah Teck", "ahteck@somewhere.com", 'm');
System.out.println(anAuthor);   // call toString()
anAuthor.setEmail("paul@nowhere.com")
System.out.println(anAuthor);
```

A class called Book is designed as follows:

- Four private instance variables: name (String), author (of the class Author you have just created, assume that each book has one and only one author), price (double), and qtyInStock (int);
- Two constructors:

- public Book (String name, Author author, double price) {...}
- public Book (String name, Author author, double price, int qtyInStock) {...}

- public methods getName(), getAuthor(), getPrice(), setPrice(), getQtyInStock(),setQtyInStock().
- toString() that returns "*'book-name' by author-name (gender) at email*". (Take note that the Author's toString() method returns "*author-name (gender) at email*" so use that!)

Write the class Book (which uses the Author class written earlier). Also write a test program called TestBook to test the constructor and public methods in the class Book. Take Note that you have to construct an instance ofAuthor before you can construct an instance of Book. E.g.,

```
Author anAuthor = new Author(......);

Book aBook = new Book("Java for dummy", anAuthor, 19.95, 1000);

//OR

// Use an anonymous instance of Author
Book anotherBook = new Book("more Java for dummy", new Author(......), 29.95, 888);
```

Take note that both Book and Author classes have a variable called name. However, it can be differentiated via the referencing instance. For a Book instance says aBook, aBook.name refers to the name of the book; whereas for an Author's instance say auAuthor, anAuthor.name refers to the name of the author. There is no need (and not recommended) to call the variables bookName and authorName.

Question 2 (30 Points)

Write a Function isPalisndrome() to check if a given string is palindrome or not. A string is a palindrome if reverse of the string is also the string itself. E.g. “tat”, “wow” , “RACECAR”, “CIVIC”. The function’s return type should be a Boolean (true if input string is a palindrome otherwise false). Also ignore case (upper/lower) of the string. i.e. “CiVic” , “Wow”, “RaceCar” all these strings are still palindrome.

In the main function, test isPalisdrome(String myString) method. Take a string as an input from console and print whether it was a palindrome or not.

Overload the same method for integer input. An integer is a palindrome if reverse of the integer is the integer itself. E.g. 1221, 1456541, 1005001. The function should take parameter of type int and should return a Boolean just like the above function.

You should use division operator (/) and remainder operator (%) to solve this question. Just remember, division operator can be used to get rid of last digit e.g. 124/10 will give you 12, and modulus operator can give you last digit e.g. 124%10 will return 4. **Do not use Java API for this problem** (like converting integer to string using java API and then using isPalindrome(String myString)). Solve it using the mathematical operators (/, %).

In the main function, test isPalisdrome(int number) method. Take an integer as an input from console and print whether it was a palindrome or not.