Programming Project 2 – Palindrome

Note: When you turn in an assignment to be graded in this class, you are making the claim that you neither gave nor received assistance on the work you turned in (except, of course, assistance from the instructor or teaching assistants).

Write a program called **Palindrome.java** that determines if a 3-digit number is a palindrome. The user will be given a choice and will enter a **1** if they would like to input the 3-digit number or they can enter a **2** to have the program randomly generate the 3-digit number. This will be done using Math.random to generate the three numbers. If a 1 is entered, the user will then enter the 3-digit number. If a 2 is entered the program will display the 3-digit number it generated. Please note that the number generated needs to be an actual 3-digit number. The program will then output whether the number is a palindrome or not. If it is, the computer displays the number with a space after it and the text, “is a palindrome”. If it is not a palindrome, then the computer displays the number with a space after it and the text, “is not a palindrome”.

The program will also be able to handle incorrect integer input. If at first the user does not enter a 1 or a 2, the program will state, “Incorrect input”. If the user is entering a three-digit number and does not enter a three-digit number, the program will also state, “Incorrect input”.

Here is a sample run:

```
Enter 1 if you would like to enter a 3-digit number.  Enter 2 if you
would like to have the computer generate it.  1

Enter the 3-digit number: 161

161 is a palindrome


Enter 1 if you would like to enter a 3-digit number.  Enter 2 if you
would like to have the computer generate it.  2

162

162 is not a palindrome
```

```
Enter 1 if you would like to enter a 3-digit number.  Enter 2 if you
would like to have the computer generate it.  5
Incorrect input

Enter 1 if you would like to enter a 3-digit number.  Enter 2 if you
would like to have the computer generate it.  1
Enter the 3-digit number: 45
Incorrect input
```

This and all program files in this course must include a comment block at the beginning (top) of the source code file that contains:

- the Java program name
- project description
- your name
- the version date
- the course number and section

The comment lines should look like this:

```
/*****************************************************************************
 * Java program name
 *****************************************************************************
 * Project description
*_____________________________________________________________________________
 * Your name
 * The version date
 * The course number and section
 *****************************************************************************/
```

Before beginning this project, you will document your algorithm as a list of steps to take you from your inputs to your outputs. This algorithm will be due on paper at the beginning of class on the day it is due. This will be graded and returned to you. It will be your responsibility to understand and correct any errors you might have with your algorithm.

Each step of your algorithm will be added as a comment block within your code. You will have the comment block right above the code that performs the actions specified.

For example, before your lines of code that ask the user for inputs, you would have a comment block that states what inputs you are requesting from the user.

You will document your tests using the form shown below.   You need to have a test case for each path through your program. I have given you two of them as examples above.

You will submit your Java source code file (Palindrome.java) to Gradescope and submit your test documentation (TestPlan.docx) by uploading the file to the Assignment link in Blackboard.

Ask questions about any part of the programming project that is not clear!

Test Plan:

Expected output – expected test results against which the output of the test is compared.

Test Name	1	2

Rubric for Programming Project 2

Item	Points
Algorithm submitted in class on time	20
Program file name correct and submitted as specified	5
Comment block at top of file with specified project information	5
Comment blocks stating the algorithm step above the code as specified including comment header block	10
Appropriate choice of variable names	5
Decisions are properly structured	10
Program layout and appearance (Coding style is clear and easily understood)	5
User prompts are in the correct order and clearly written	5
Handles incorrect integer input	5
Output is correct	20
Documentation of test plan submitted	10
Total	**100**