Name __________________
Date ___________________

AP Computer Science A
Unit 4 For Loop Lab*

Directions: Write the code for all of the following programs in one main method.

1. **Nursery Rhyme 1**
   Use a for loop to print out all of the lyrics of Ten Little Monkeys Jumping on a Bed. You can use digits in your lyric output.

2. **Nursery Rhyme 2**
   Use a for loop to print out all the lyrics of 99 bottles of milk on the wall. You can use digits in your lyric output.

3. **Number of a Letter**
   Have the user input a String (word, phrase, sentence, paragraph, etc) and indicate a letter of interest. Use a for loop to determine and print out how many times that letter appears in the String.

4. **Jump Rope for Heart**
   The American Heart Association is having a Jump Rope for Heart event in which participants donate money for each jump they make. Write a program that asks the user how many participants there are. Then, use a for loop to keep track of the total donations. Ask for each participant's donation rate per jump. Then ask how many jumps each person makes. Calculate the amount of money earned per person and add to the total donations. After all participant information has been entered and accounted for, print out the total amount earned by the Jump Rope for Heart Program.

*Please note that these will be graded along with the other Unit 4 Exercises for one overall Lab Grade.

For Loop Lab Rubric

	3	2	1	0
Nursery Rhyme 1	Program uses for loop to correctly print the lyrics of the rhyme	Program uses for loop to print the lyrics of the rhyme but with error	Program prints the lyrics without using a for loop	Program does not print the lyrics of the rhyme
Nursery Rhyme 2	Program uses for loop to correctly print the lyrics of the rhyme	Program uses for loop to print the lyrics of the rhyme but with error	Program prints the lyrics without using a for loop	Program does not print the lyrics of the rhyme
Number of a Letter – Input	Program allows user to input appropriate information throughout the program	Program allows for user to input information but with error	Program partially allows for user to input information	Program does not allow user to input information
Number of a Letter – for loop	Program uses for loop correctly to account for all indexes of the String	Program uses for loop but has a one-off error	Program uses for loop but with errors	Program does not use a for loop
Number of a Letter – comp/output	Program correctly tallies the number of the indicated letter and prints out the correct total count	Program correctly tallies and prints the total number of the indicated letter but with one error	Program correctly tallies and prints the total number of the indicated letter but with errors	Program does not tally or print the total number of the indicated letter
Jump Rope – input	Program allows user to input appropriate information throughout the program	Program allows for user to input information but with error	Program partially allows for user to input information	Program does not allow user to input information
Jump Rope – for loop	Program uses for loop correctly to account for all participants	Program uses for loop but has a one-off error	Program uses for loop but with errors	Program does not use a for loop
Jump Rope – calc/output	Program correctly computes and prints out the donations	Program computes and prints donations with error	Program computes the donations but does not print the total donations	Program does not compute or print the donations
Style	Classes follow Java style guidelines and include appropriate variable names, comments, etc.	Classes follow Java style guidelines and include 1-2 questionable variable names, comments, etc.	Classes follow Java style guidelines and include multiple questionable variable names, comments, etc.	Classes are missing comments and/or have inappropriate variable names.

*Please note that these will be graded along with the other Unit 4 Exercises for one overall Lab Grade.

Name:
Date:

AP Computer Science A
Unit 4 While Loop Lab*

1. **Compute Sum of Digits**
   Create a program that will prompt the user to type in a positive integer (you should include code to ensure that the input is positive and prompt the user for a new number if it is not) and computes the sum of the digits of the that integer. Your output should look similar to the following sample output:

```
Type an integer: 827104
Digit sum: 22
```

2. **Guessing Game 1**
   Create a program that will use a while loop to play a guessing game. Your game should have (1) a variable that holds the winning number between the values of 1 and 100 inclusive (you should use Math.random() to generate this number), and (2) a variable that will hold the user's guess. Incorporate if statements to guide the user towards the right answer (e.g. let the know if their guess is too high or too low). Your output should look similar to the following sample output:

```
I have chosen a number between 1 and 100. Try to guess it.
Your guess: 5
That is too low. Guess again.
Your guess: 40
That is too high. Guess again.
Your guess: 8
That is too low. Guess again.
Your guess: 11
That's right! You're a good guesser.
```

3. **Guessing Game 2**
   Modify your program from Guessing Game 1 (copy and paste the code so you do not need to retype everything!). All the conditions from the previous program still hold, but now, the user has a limit on the number of guesses he or she is allowed. If the user is not able to guess the number within 6 tries, the game is over and appropriate output should be printed.

*Please note that these will be graded along with the other Unit 4 Labs for one overall Lab Grade.

4. **Palindrome**

   Write a method

   public boolean isPalindrome (String word)

   that tests whether word is a palindrome (the same when read forward or backward, as in "madam"). Test isPalindrome using a test program.

   Upgrade isPalindrome so that it can hand any phrase (as in "Madam, I'm Adam"). In testing for a palindrome, disregard all spaces, punctuation marks, apostrophes, and other non-alphanumeric characters and consider lower- and uppercase letters the same. Do not count an empty string as a palindrome. (Remember the character class has static methods boolean isLetterOrDigit (ch) and char toUpperCase(ch)!)

*Please note that these will be graded along with the other Unit 4 Labs for one overall Lab Grade.

Rubric

	3	2	1	0
Sum – Input	Program prompts user for input and has a while loop to check non-negativity condition	Program prompts user for input and has a check for non-negativity condition but has an error	Program prompts user for input but does not check for non-negativity condition	Program does not allow for user input.
Sum – Calculation	Program uses while loop to correctly calculate the sum of the digits	Program uses while loop but incorrectly calculates the sum of the digits	Program attempts to calculate sum without the use of a while loop	Program does not attempt to calculate the sum of the digits
Sum – Output	Program prints the integer and the sum of its digits with context	Program prints the integer and the sum of its digits with some context	Program prints the integer and the sum of its digits with no context.	Program does not print the integer and the sum of its digits.
GG 1 – winning #	Program properly uses the Math.Random() method to generate a winning number between 1-100	Program uses Math.Random() to generate a winning number but with error	Program has a constant winning number	Program does not have a winning number assigned
GG 1 – user guess	Program prompts user for guesses throughout the program	Program prompts user for guesses but does so with error in the while loops	Program only prompts user for an initial guess	Program does not prompt user for guesses
GG 1 – high/low	Program correctly guides user with too high/too low using appropriate if statements	Program guides user towards the winning number but if statements have an error	Program guides user towards the winning number incorrectly	Program does not guide user towards the winning number
GG 2 – 2nd condition	Program correctly implements second condition for guessing	Program implements second condition for guessing but with an error	Program implements second condition for guessing but with multiple errors	Program does not implement second condition for guessing
Palindrome	Program correctly tests whether a word or phrase is a palindrome	Program tests whether a word or phrase is a palindrome with 1 error	Program tests whether a word or phrase is a palindrome with multiple errors	Program does not test whether a word or phrase is a palindrome
Upgraded Palindrome - Characters	Program correctly disregards all characters that are not letters.	Program disregards characters that are not letters with 1 error	Program disregards characters that are not letters with multiple errors	Program does not disregard characters that are not letters
Upgraded Palindrome - test	Program correctly tests whether a word or phrase is a palindrome	Program tests whether a word or phrase is a palindrome with 1 error	Program tests whether a word or phrase is a palindrome with multiple errors	Program does not test whether a word or phrase is a palindrome

*Please note that these will be graded along with the other Unit 4 Labs for one overall Lab Grade.

Upgraded Palindrome – empty	Empty string condition +1			No empty string condition +0
Style	Classes follow Java style guidelines and include appropriate variable names, comments, etc.	Classes follow Java style guidelines and include 1-2 questionable variable names, comments, etc.	Classes follow Java style guidelines and include multiple questionable variable names, comments, etc.	Classes are missing comments and/or have inappropriate variable names.

*Please note that these will be graded along with the other Unit 4 Labs for one overall Lab Grade.

Name:
Date:

AP Computer Science A
ATM Machine Lab

We are going to create a program that simulates an ATM machine. Features of our ATM machine include:

- User/PIN security—users must enter in a valid user/pin combination to access their account
- Balance information—users may access their account balance
- Check deposit—users may deposit multiple checks at one time
- Money withdrawal—users may withdraw money in increments of $20

Account Security:

1. In order to get started, we will first need to create a user/pin combination that allows users to login. We will do that by creating a couple instance, constant variables at the top of our program, directly under where we create our Scanner in variable. You should have your own user/pin combination.

   ```
   private static final String USERNAME = “user”;
   private static final int PIN = 1234;
   ```

   We will then be able to use these variables throughout our program, specifically when we prompt the user for their credentials (and make sure they match the finals) before giving them access to their account.

2. Next we will prompt the user for a username and pin. We will need to check if the username/pin that are entered match the final variables. If they enter the correct password, they should proceed to account management. Give users 3 chances to enter in the correct credentials. Users should be informed that their account has been locked temporarily and the program should end if they do not give the correct credentials.

Account Management:

3. Users should be prompted with a menu that allows them to 1) Check account balance 2) Deposit Checks 3) Withdraw Money 4) Logout. The menu should repeat after each choice until the user chooses to logout. Ensure that the user input for a menu option is valid. If they choose an invalid option, tell them.

4. **Check account balance**—Display the user’s account balance. (You will need to create another instance variable to keep track of this.)

5. **Deposit Checks**—Ask the user how many checks they would like to deposit and then prompt for the amount of each check (each check may be a different amount, so prompt for each check's amount). Add the appropriate amount to the account balance.

6. **Withdraw Money**—Ask the user how much money they would like to withdraw and the subtract that amount from their account. Do not allow the withdrawal to exceed the account balance. Only allow withdrawals in increments of $20. If the user chooses an amount that is not a multiple of 20, they should receive an error message and be prompted for a valid amount to withdraw.

Rubric

	3	2	1	0
Global constants	Program correctly declares and initializes instance constants for user name and pin	Program declares or initializes instance constants but has an error	Program declares or initializes instance constants incorrectly	Program does not declare and initialize instance constants
Account security	Program correctly checks credentials and gives user a maximum of 3 attempts	Program correctly checks credentials but does not give user a limit on attempts	Program checks credentials but with error	Program does not check user credentials
Menu	Program displays menu, ensures valid user input, and reprints until user chooses to log out	Program displays menu and reprints until user chooses to log out but does not ensure valid input	Program attempts all functions related to the menu but with error(s)	Program does not display a menu
Check balance	Program correctly declares instance variable and checks and displays balance	Program does not declare instance variable but checks and displays balance	Program declares a variable, checks and displays balance but with error	Program does not declare a variable, check or display the balance
Deposit checks	Program allows user to input desired number of checks, correctly gets value of each check, and correctly adds value to balance	Program allows user to input desired number and value of checks but does not calculate the account balance correctly	Programs allows user input but does has errors in getting all necessary input and/or does not calculate the account balance	Program does not allow user to deposit checks
Withdraw money	Program allows user to withdraw money in increments of $20 only and adjusts the account balance correctly	Program allows user to withdraw money with no $20 increment restriction and adjusts the account balance correctly	Program allows user to withdraw money but adjusts the account balance incorrectly	Program does not allow user to withdraw money
Style	Classes follow Java style guidelines and include appropriate variable names, comments, etc.	Classes follow Java style guidelines and include 1-2 questionable variable names, comments, etc.	Classes follow Java style guidelines and include multiple questionable variable names, comments, etc.	Classes are missing comments and/or have inappropriate variable names.

Name ________________________
Date __________________________

AP Computer Science A
Unit 4 Nested Loop Lab

Use one main method for all the prompts in this exercise. Use comments to show the start of each question. At the top of your main method, you should prompt the user for a positive integer to determine the number of rows for each of the patterns. If the user does not input a positive integer, allow the user to re-enter information until all requirements are met. All of the sample output below are for a user input of 4.

1. **Stars1**
   Given the user input, n, print a nxn square of *.

```
****
****
****
****
```

2. **Stars2**
   Given the user input, n, print a triangle of * where the first row has 1 star and the last row has n stars.

```
*
**
***
****
```

3. **Stars3**
   Given the user input, n, print a triangle of * where the first row has 1 star and the last row has n stars but the stars are aligned to the right.

```
   *
  **
 ***
****
```

4. **Numbers1**

   Given the user input, n, print a triangle of digits where the first row has one 1, the second has two 2’s, …, and the last row has n n’s.

   ```
   1
   22
   333
   4444
   ```

5. **Numbers2**

   Given the user input, n, print a triangle of digits where the first row prints 1, the second prints 12,…, and the last row prints the values 1 through n.

   ```
   1
   12
   123
   1234
   ```

6. **Numbers 3**

   Given the user input, n, print a triangle of all the digits starting at 1 and increasing by 1, where each row has the same number of digits as the row number

   ```
   1
   2 3
   4 5 6
   7 8 9 10
   ```

7. Bonus:

   Given the user input, n, print a triangle of all stars in a pyramid formation.

   ```
      *
     **
    ***
   ****
   ```

Rubric

	3	2	1	0
Input	Program allows user to input appropriate information and allows user to re-enter information as necessary	Program allows for user to input information and checks for requirements but only allows user to re-enter information once	Program allows for user to input information but does not check for requirements	Program does not allow user to input information
Stars1	Program uses nested loops to correctly print the pattern described	Program uses nested for loops to print the pattern described but with error	Program prints the pattern without using nested loops	Program does not print the pattern described
Stars2	Program uses nested loops to correctly print the pattern described	Program uses nested for loops to print the pattern described but with error	Program prints the pattern without using nested loops	Program does not print the pattern described
Stars3	Program uses nested loops to correctly print the pattern described	Program uses nested for loops to print the pattern described but with error	Program prints the pattern without using nested loops	Program does not print the pattern described
Numbers1	Program uses nested loops to correctly print the pattern described	Program uses nested for loops to print the pattern described but with error	Program prints the pattern without using nested loops	Program does not print the pattern described
Numbers2	Program uses nested loops to correctly print the pattern described	Program uses nested for loops to print the pattern described but with error	Program prints the pattern without using nested loops	Program does not print the pattern described
Numbers3	Program uses nested loops to correctly print the pattern described	Program uses nested for loops to print the pattern described but with error	Program prints the pattern without using nested loops	Program does not print the pattern described
Style	Classes follow Java style guidelines and include appropriate variable names, comments, etc.	Classes follow Java style guidelines and include 1-2 questionable variable names, comments, etc.	Classes follow Java style guidelines and include multiple questionable variable names, comments, etc.	Classes are missing comments and/or have inappropriate variable names.
Bonus	Program uses nested loops to correctly print the pattern described			Program does not print the pattern described