- To "capitalize" a string means to change the first letter of each word in the string to upper case (if it is not already upper case). For example, a capitalized version of "Now is the time to act!" is "Now Is The Time To Act!". Write a subroutine named printCapitalized that will print a capitalized version of a string to standard output. The string to be printed should be a parameter to the subroutine.

```
   static void printCapitalized( String str ) {
  char ch;       // One of the characters in str.
  char prevCh;   // The character that comes before ch in the string.
  int i;         // A position in str, from 0 to str.length()-1.
  prevCh = '.';  // Prime the loop with any non-letter character.
  for ( i = 0;  i < str.length();  i++ ) {
    ch = str.charAt(i);
    if ( Character.isLetter(ch)  &&  ! Character.isLetter(prevCh) )
       System.out.print( Character.toUpperCase(ch) );
    else
       System.out.print( ch );
    prevCh = ch;  // prevCh for next iteration is ch.
  }
  System.out.println();
}
```

- The hexadecimal digits are the ordinary, base-10 digits '0' through '9' plus the letters 'A' through 'F'. In the hexadecimal system, these digits represent the values 0 through 15, respectively. Write a function named hexValue that uses a switch statement to find the hexadecimal value of a given character. The character is a parameter to the function, and its hexadecimal value is the return value of the function. You should count lower case letters 'a' through 'f' as having the same value as the corresponding upper case letters. If the parameter is not one of the legal hexadecimal digits, return -1 as the value of the function.

```
public static void main(String[] args) {

    String hex;  // Input from user, containing a hexadecimal number.

    long dec;    // Decimal (base-10) equivalent of hexadecimal number.

    int i;       // A position in hex, from 0 to hex.length()-1.

    TextIO.put("Enter a hexadecimal number: ");

    hex = TextIO.getlnWord();

    dec = 0;

    for ( i = 0; i < hex.length(); i++ ) {

      int digit = hexValue( hex.charAt(i) );
```

```
    if (digit == -1) {
       TextIO.putln("Error:  Input is not a hexadecimal number.");
       return;
    }
    dec = 16*dec + digit;
  }
  TextIO.putln("Base-10 value:  " + dec);
}  // end main()

static int hexValue(char ch) {
    // Returns the hexadecimal value of ch, or returns
    // -1 if ch is not one of the hexadecimal digits.
  switch (ch) {
    case '0':
      return 0;
    case '1':
      return 1;
    case '2':
      return 2;
    case '3':
      return 3;
    case '4':
      return 4;
    case '5':
      return 5;
```

```
case '6':
  return 6;
case '7':
  return 7;
case '8':
  return 8;
case '9':
  return 9;
case 'a':
case 'A':
  return 10;
case 'b':
case 'B':
  return 11;
case 'c':
case 'C':
  return 12;
case 'd':
case 'D':
  return 13;
case 'e':
case 'E':
  return 14;
case 'f':
case 'F':
```

```
        return 15;
      default:
        return -1;
    }
}
```

- Write a program called ReverseString, which prompts user for a String, and prints the reverse of the String. The output shall look like:

  Enter a String: abcdef

  The reverse of the String "abcdef" is "fedcba".

- Write a program called CheckVowelsDigits, which prompts the user for a String, counts the number of vowels (a, e, i, o, u, A, E, I, O, U) and digits (0-9) contained in the string, and prints the counts and the percentages (with 2 decimal digits). For example,

```
Enter a String: testing12345
Number of vowels: 2 (16.67%)
Number of digits: 5 (41.67%)
```

- Copying charactor array to string
- String compare
- Comparing StringBuffer with a String
- Getting byte array from a string
- String split function
- Extract char array from string
- Replace characters in a string
- Changing string case
- Trim spaces in string
- How to format a string?
- How to match a string with regular expression?
- How to remove multiple space in a string?
- How to remove non-ascii characters from a string?
- How to remove html tags from a string?
- How to get line count from a string?
- test if a given string contains the specified sequence of char values.
- to check whether a given string ends with the contents of another string.
- to find whether a region in the current string matches a region in another string.
- A word that reads the same backward as forward is called a palindrome, e.g., "mom", "dad", "racecar", "madam", and "Radar" (case-insensitive). Write a program called TestPalindromicWord, that prompts user for a word and prints ""xxx" is|is not a palindrome".

- How to Print duplicate characters from String?
- How to check if two Strings are anagrams of each other?
  Two strings are anagrams if they are written using the same exact letters, ignoring space, punctuation and capitalization. Each letter should have the same count in both strings. For example, Army and Mary are anagram of each other.

- How to program to print first non repeated character from String?
- How to check if a String contains only digits?
- How to count occurrence of a given character in String?
- How to find all permutations of String?

- Given a string, return true if "bad" appears starting at index 0 or 1 in the string, such as with "badxxx" or "xbadxx" but not "xxbadxx". The string may be any length, including 0. Note: use .equals() to compare 2 strings.

  hasBad("badxx") → true
  hasBad("xbadxx") → true
  hasBad("xxbadxx") → false
- Given a string of odd length, return the string length 3 from its middle, so "Candy" yields "and". The string length will be at least 3.
  middleThree("Candy") → "and"
  middleThree("and") → "and"
  middleThree("solving") → "lvi"
- Given a string and an index, return a string length 2 starting at the given index. If the index is too big or too small to define a string length 2, use the first 2 chars. The string length will be at least 2.
  twoChar("java", 0) → "ja"
  twoChar("java", 2) → "va"
  twoChar("java", 3) → "ja"

- Given a string and an int n, return a string made of the first and last n chars from the string. The string length will be at least n.
  nTwice("Hello", 2) → "Helo"
  nTwice("Chocolate", 3) → "Choate"
  nTwice("Chocolate", 1) → "Ce"
- Given a string, return true if it ends in "ly".
  endsLy("oddly") → true
  endsLy("y") → false
  endsLy("oddy") → false
- Given a string, return a version without both the first and last char of the string. The string may be any length, including 0.