**AP CS A – Week 9 Exercises**

**Exercise 43: Sum of Numbers**

Create the method **sum** that calculates the sum of numbers the method receives as parameters.

Place the method in the following program body:

**public** **static** int **sum**(int number1, int number2, int number3, int number4) {

*// write program code here*

*// remember that the method needs a return in the end*

}

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

int answer = **sum**(4, 3, 6, 1);

System.out.**println**("sum: " + answer);

}

Example output:

sum: 14

**Note:** if an exercise involves a method *returning* something, it means that the return type needs to be defined for the method, and that the method needs to return a value of that type using the return command. In this case, the method does not print (or use the command System.out.println(..)), the method caller handles printing, here, the main program.

**Exercise 47: Length of a name**

Create a program that asks for the user's name and says how many characters the name contains.

Type your name: Paul

Number of characters: 4

Type your name: Catherine

Number of characters: 9

**Note!** Your program should be structured so that you put the calculating of the name length in it's own method: public static int calculateCharacters(String text). The tests will be testing both the method calculateCharacters and the program overall.

**Exercise 49: Last Character**

Create a program that asks for the user's name and gives the last character.

Type your name: Paul

Last character: l

Type your name: Catherine

Last character: e

**Note!** Your program should be structured so that you put the search for the last character in its own method: public static char lastCharacter(String text). The tests will be testing both the method lastCharacter and the program overall.

**Exercise 55: A word inside a word**

Create a program that asks the user for two words. Then the program tells if the second word is included in the first word. Use String method indexOf in your program.

Type the first word: glitter

Type the second word: litter

The word 'litter' is found in the word 'glitter'.

Type the first word: glitter

Type the second word: clean

The word 'clean' is not found in the word 'glitter'.

**Note:** Make your program outputs (prints) match exactly the example above!

**Exercise 56: Reversing Text (BONUS)**

Create the method **reverse** that puts the given string in reversed order. Use the following program body for the method:

**public** **static** String **reverse**(String text) {

*// write your code here*

}

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

System.out.**print**("Type in your text: ");

String text = reader.**nextLine**();

System.out.**println**("In reverse order: " + **reverse**(text));

}

**Hint:** you probably need to build the reversed string character by character in your method. You can use a String-type variable as a helper during the building process. In the beginning, the helper variable should have an empty string of characters as a value. After this, new characters are added to the string one by one.

String help = "";

// ...

// adding a character to the help variable

help = help + character;

Program output:

Type a text: example

elpmaxe