**44-542 Object Oriented Programming**

**Recursion**

**Objective** This worksheet will give you practice in working with recursive methods.

Trace the code and write it below each question for following methods.

**Example:**

Factorial may be defined recursively:

**fact(n) = n \* fact(n - 1) if n > 0;**

**fact(0) = 1**

Using this definition

**fact(2) = 2 \* fact(1) = 2 \* (1 \* fact(0)) = 2 \* (1 \* 1) =**

**2 \* 1 = 2**

Similarly, find the value returned by each of the following methods and show each step in the space provided.

**public static int mystery(int numIn) {**

**if (numIn == 0) {**

**return 0;**

**} else {**

**if ((numIn % 10) % 2 == 0) {**

**return numIn % 10 + mystery(numIn / 10);**

**} else {**

**return mystery(numIn / 10);**

**}**

**}**

**}**

* 1. **mystery(8342)**

|  |
| --- |
| 2 + mystery(834)  2 + 4 + mystery(83)  2 + 4 + mystery(8)  2 + 4 + 8 + mystery(0)  2 + 4 + 8  14 |

* 1. **mystery(1357)**

|  |
| --- |
| mystery(135)  mystery(13)  mystery(1)  0 |

**public static int mystery(int numIn) {**

**if (numIn == 0) {**

**return 1;**

**} else {**

**if ((numIn / 10) % 2 == 0) {**

**return numIn / 10 - mystery(numIn / 10);**

**} else {**

**return mystery(numIn / 10);**

**}**

**}**

**}**

* 1. **mystery(8342)**

|  |
| --- |
| 834 - mystery(834)  834 - mystery(83)  834 - ( 8 - mystery(8))  834 - ( 8 - (0-mystery(0)))  834 - ( 8 - (0 - 1))  834 - (8 + 1)  834 - 9  825 |

* 1. **mystery(1357)**

|  |
| --- |
| mystery(135)  mystery(13)  mystery(1)  0 - mystery(0)  0 - 1  -1 |



**public class RecursionExercise {**

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

**System.out.println(mystery("Hello"));**

**System.out.println(**

**mystery("Thanks Giving Break is soon!"));**

**}**

**public static String mystery(String str) {**

**if (!str.contains(" ")) {**

**return rw(str);**

**} else {**

**return rw(str.substring(0, str.indexOf(" "))) + " "**

**+ mystery(str.substring(str.indexOf(" ") + 1));**

**}**

**}**

**public static String rw(String str) {**

**String rStr = "";**

**for (int i = str.length() - 1; i >= 0; i--) {**

**rStr += str.charAt(i);**

**}**

**return rStr;**

**}**

**}**

|  |
| --- |
| mystery(“Hello”)  rw(“Hello”)  olleH  mystery("Thanks Giving Break is soon!")  rw(“Thanks”) + “ “ + mystery(“Giving Break is soon!”)  rw(“Thanks”) + “ “ + rw(“Giving”) + mystery(“Break is soon!”)  rw(“Thanks”) + “ “ + rw(“Giving”) + rw(“Break”) + mystery(“is soon!”)  rw(“Thanks”) + “ “ + rw(“Giving”) + rw(“Break”) + rw(“is”) + mystery(“soon!”)  rw(“Thanks”) + “ “ + rw(“Giving”) + rw(“Break”) + rw(“is”) + rw(“soon!”)  sknahT gniviG kaerB si !noos |

4)

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

**System.out.println(mystery("Bearcats"));**

**System.out.println(mystery("Northwest Bearcats Defeat Fort Hays"));**

**}**

**public static String mystery(String str) {**

**if (!str.contains(" ")) {**

**return " DONE";**

**} else {**

**return str.substring(str.indexOf(" ") + 1, str.indexOf(" ") + 2)**

**+ mystery(str.substring(str.indexOf(" ") + 1));**

**}**

**}**

|  |
| --- |
| DONE    B + mystery(“Bearcats Defeat Fort Hays”)  B + D + mystery(“Defeat Fort Hays”)  B + D + F + mystery(“Fort Hays”)  B + D + F + H + mystery(“Hays”)  B + D + F + H + DONE  BDFH DONE |