**CS 1301 – Intro to CS**

**Exercise – Methods**

**A set of problems were taken from https://codingbat.com**

1. **We have a number of bunnies and each bunny has two big floppy ears. We want to compute the total number of ears across all the bunnies (Use methods and loops, no multiplication).**

bunnyEars(0) → 0  
bunnyEars(1) → 2  
bunnyEars(2) → 4

public class Main{

public static int bunnyEars(int n){

int ears= 0;

while(n>0){

ears +=2;

n--;

}

return ears;

}

Public static void main(string[]args){

System.out.print(bunnyEars(0));

System.out.print(bunnyEars(1));

System.out.print(bunnyEars(2));

}

}

1. **We have bunnies standing in a line, numbered 1, 2, ... The odd bunnies (1, 3, ..) have the normal 2 ears. The even bunnies (2, 4, ..) we'll say have 3 ears, because they each have a raised foot. Return the number of "ears" in the bunny line 1, 2, ... n (Use methods and loops, no multiplication).**

bunnyEars2(0) → 0

bunnyEars2(1) → 2

bunnyEars2(2) → 5

public class Main{

public static int bunnyEars2(int n){

int ears = 0;

for(int i = 1; i<=n; i++){

if(i%2= = 1){

ears+=2;

}else{

ears+=3;

}

Return ears;

}

Public static void main(String[]args){

System.out.println(bunnyEars2(0));

System.out.println(bunnyEars2(1));

System.out.println(bunnyEars2(2));

}

1. **Given a non-negative int n, return the sum of its digits (use loops). Note that mod (%) by 10 yields the rightmost digit (126 % 10 is 6), while divide (/) by 10 removes the rightmost digit (126 / 10 is 12).**

sumDigits(126) → 9

sumDigits(49) → 13

sumDigits(12) → 3

public static int sumOfDigits(int n){

if (n<=0){

return 0;

} else {

return (n%10) + sumOfDigits(n/10);

}

1. **Given a non-negative int n, return the count of the occurrences of 7 as a digit, so for example 717 yields 2. (use loops). Note that mod (%) by 10 yields the rightmost digit (126 % 10 is 6), while divide (/) by 10 removes the rightmost digit (126 / 10 is 12).**

count7(717) → 2

count7(7) → 1

count7(123) → 0

public static int count7(int n){

int c=0;

if(7>n){

return 0;

}else{

if (7 ==n%10){

c = 1;

} else {

c = 0;

}

}

return c + count7(n/10);

}

1. **Given a string, compute (using loops) and return a new string where all the lowercase 'x' chars have been changed to 'y' chars.**

Public static String changeXY(String n){

String x = “ ”;

For(int i=0;i<s.length; i++){

x+=y;

} else{

X +=s.charAt(i);

}

Return x;

}

1. **Given a non-negative int n, compute (using loops) the count of the occurrences of 8 as a digit, except that an 8 with another 8 immediately to its left counts double, so 8818 yields 4. Note that mod (%) by 10 yields the rightmost digit (126 % 10 is 6), while divide (/) by 10 removes the rightmost digit (126 / 10 is 12).**

count8(8) → 1

count8(818) → 2

count8(8818) → 4

public int count8(int n){

if (n<1){

return 0;

}

If (n% 10 == 8 && (n/10)%10 == 8){

return 2 + count8(n/10);

} Else if (n % 10 == 8) { return 1 + count8(n/10);

} else { return count8(n/10);

}

1. **Write a method called printMin that receives three integers as parameters and prints the smallest of the three**

Public static void printMin(int a, int b, int c){

Int min= a;

If( b<min){

Min =b;

}

If(c<min){

Min = c;

}

System.out.println(min);

}

1. **Write a method called printComputeAve that receives three integers as parameters and prints their average**

public static void printComputeAve(int a, int b, int c){

int average = (a+b+c)/3;

System.out.println(average);

}

1. **Write a method called printNumVowels that receives a String as a parameter and prints the count of all vowels in the string**

Public static void printNumVowels(String s){

Int numVowels = 0;

Char c;

For(int i=0; i<s.length();i++){

c = s.charAt(i);

if( c==’a’ || c==’e’||c==’i’||c==’o’||c==’u’){

++numVowels;

}

}

System.out.println(numVowels);

}

1. **Write a method called printIsValidPassword that receives a String as input and prints *true* if and only if the provided string is a valid password. A valid password is a string with at least 8 characters. If this is too easy, add the following constraints:**
   1. **A valid password consists of only letters and digits (0-9)**
   2. **A valid password must contain at least two digits (0-9)**

Public static boolean printIsValidPassword(string in){

If(s.length>=8){

Return true;

System.out.println(“true”);

} else {

Return false;

System.out.println(“false”);

}

}

1. **Write a method called printIsMultiple that receives a pair of integers and prints true if the second is a multiple of the first; false, otherwise**

public class printsIsMultiple{

public static boolean printsIsMultiple(int x, int y){

Boolean a = false;

If(x%y==0){

a = true;

}

Return a;

}

Public static void main(String[]args){

Scanner scanner = new Scanner(System.in)

Int x = scanner.nextInt();

Int y = scanner.nextInt();

Boolean a = printsIsMultiple(x,y);

System.out.println(a);

1. **Write a method called printComputeArea that receives the radius of a circle as an argument and prints its area**

public static void main (String[]args){

Double radius=7.0;

printComputeArea(radius);

}

public static void printComputeArea(double radius){

if(radius<0){

System.out.println(“Try again with a positive number.”);

}else{

Double area=3.14\*radius\*radius;

System.out.println(area);

}

}

}