**Northeastern Illinois University**

**CS-200: Programming I**

**PLTL: Loops**

**Problem 1**

* Write a program that has the class name Problem1 and that has the main method.
* Write a program that simulates flipping a coin one million times and displays the number of heads and tails.
* Remember that you can represent a flip that results in heads using the number 0 and a

flip that results in tails using the number 1 (or vice-versa!).

* Several sample runs are provided for you below. Your output must be formatted **exactly**

like the sample runs below. Note that while your output must be formatted as below, you will not get the same results as this uses random numbers.

|  |
| --- |
| Heads: 500178  Tails: 499822  500178 Heads + 499822 Tails is 1000000 flips |

**Problem 2**

* Write a program that has the class name Problem2 and that has the main method.
* In the main method, prompt the user to enter positive integers until the user enters either

a value that is less than or equal to zero **or** when the number entered is less than or equal

to the number entered previously.

* A step is a positive difference between an entered integer and the value entered prior

to that integer. Calculate the number of big steps (differences of more than 5) and the

number of small steps (differences of less than or equal to 5).

* Print out the big steps and the small steps and the number that caused the program to

end.

* Several sample runs are provided for you below. Your output must be formatted **exactly**

like the sample runs below.

|  |
| --- |
| Enter a positive integer: 4  Enter a positive integer: 12  Enter a positive integer: 13  Enter a positive integer: 15  Enter a positive integer: 21  Enter a positive integer: 26  Enter a positive integer: 21  Big steps: 2  Small steps: 3  Ending value: 21 |

|  |
| --- |
| Enter a positive integer: 1  Enter a positive integer: -3  Big steps: 0  Small steps: 0  Ending value: -3 |

|  |
| --- |
| Enter a positive integer: -10  Big steps: 0  Small steps: 0  Ending value: -10 |

|  |
| --- |
| Enter a positive integer: 45  Enter a positive integer: 62  Enter a positive integer: 0  Big steps: 1  Small steps: 0  Ending value: 0 |

|  |
| --- |
| Enter a positive integer: 71  Enter a positive integer: 72  Enter a positive integer: 72  Big steps: 0  Small steps: 1  Ending value: 72 |

**Problem3**

* Write a program that has the class name Problem3 and that has the main method.
* In the main method prompt the user to enter up to 5 positive integers between 1 and 20

(inclusive).

* The program should count and add the positive values.
* Stop prompting the user to enter values if the user enters 5 numbers or a non-positive

number.

* Print out the total number of positive integers and their product.
* Several sample runs are provided for you below. Your output must be formatted **exactly**

like the sample runs below.

|  |
| --- |
| Enter a positive integer: 12  Enter a positive integer: 17  Enter a positive integer: 2  Enter a positive integer: 0  Number of positive numbers is: 3  The product of positive numbers is: 408 |

|  |
| --- |
| Enter a positive integer: 6  Enter a positive integer: 5  Enter a positive integer: 7  Enter a positive integer: 3  Enter a positive integer: 19  Number of positive numbers is: 5  The product of positive numbers is: 11970 |

|  |
| --- |
| Enter a positive integer: 356  Enter a positive integer: -6  Number of positive numbers is: 1  The product of positive numbers is: 356 |

|  |
| --- |
| Enter a positive integer: 0  Number of positive numbers is: 0  The product of positive numbers is: 0 |

**Problem4**

* Write a program that has the class name Problem4 and that has the main method.
* In the main method prompt the user to enter a positive integer. You can assume that they will enter a positive value.
* Calculate the sum of the digits in the number and print out the value.
* If the sum is greater than or equal to 10, then reduce the sum again to a value less than 10 by adding the individual digits. Continue to do this until your sum is less than 10.
* Several sample runs are provided for you below. Your output must be formatted **exactly**

like the sample runs below.

|  |
| --- |
| Enter a positive integer: 34849  Digit sum is 28  Digit sum is 10  Digit sum is 1 |

|  |
| --- |
| Enter a positive integer: 99993  Digit sum is 39  Digit sum is 12  Digit sum is 3 |

|  |
| --- |
| Enter a positive integer: 356  Digit sum is 14  Digit sum is 5 |

|  |
| --- |
| Enter a positive integer: 100005  Digit sum is 6 |

**Problem 5**

* Write a program that has the class name Problem5 and that has the main method. The

main method should take user input for one integer.

* The create another method called isPrime that is void and takes one integer as a parameter.
* Write a third method named check that returns a boolean and takes an integer as a

parameter.

* The program should print out true if the number entered by the user is prime and false

if otherwise.

* Several sample runs are provided for you below. Your output must be formatted **exactly**

like the sample runs below. Note that while your output must be formatted as below, you

will not get the same results as this uses random numbers.

|  |
| --- |
| Enter n: 9  false |

|  |
| --- |
| Enter n: 5  true |

**Problem 6**

* Write a program that has the class name Problem6 and that has the main method.
* Ask the user to enter integers until 3 consecutive increasing integers are enter.
* The program should display the sum of all the integers entered.
* Several sample runs are provided for you below. Your output must be formatted **exactly** like the sample runs below.

|  |
| --- |
| Enter an int: 3  Enter an int: 4  Enter an int: 5  12 |

|  |
| --- |
| Enter an int: 20  Enter an int: -2  Enter an int: -1  Enter an int: 0  17 |

|  |
| --- |
| Enter an int: 1  Enter an int: 7  Enter an int: 8  Enter an int: 2  Enter an int: 9  Enter an int: 10  Enter an int: 11  48 |