**Northeastern Illinois University**

**CS-200: Programming I**

**PLTL: Math Selection Statements**

**Problem 1 (SameLastDigit)**

* Write a program that has the class name SameLastDigit and that has the main method.
* The program should ask the user to enter two integers.
* The program should determine whether the two integers have the same last digit. If they do, print out Same last digit. Otherwise, print out Different last digits.
* Several sample runs are provided for you below. Format your output to match the sample output.

|  |  |
| --- | --- |
| First number: 1092  Second number: 62  Same last digit | First number: 35  Second number: 83213  Different last digits |

**Problem 2 (Makes10)**

* Write a program that has the class name Makes10 and that has the main method.

The program should ask the user to enter two integers.

* The program should determine whether the two integers sum up to 10 or if one of them is 10. Based on whether the integers sum up to 10 or one of them is 10, print out the appropriate message.
* Several sample runs are provided for you below. Format your output to match the sample output.

|  |  |
| --- | --- |
| First number: -10  Second number: 20  -10 + 20 sums up to 10  Neither number is 10 | First number: 10  Second number: 33  Does not sum up to 10  One number is 10 |

-

|  |  |
| --- | --- |
| First number: 10  Second number: 0  10 + 0 sums up to 10  One number is 10 | First number: 3  Second number: 12  Does not sum up to 10  Neither number is 10 |

**Problem 3 (Distance)**

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

The program should ask the user to enter four numbers representing the x and y coordinates of two points.

* The program should calculate and output the distance between the two points using the following formula:
* Several sample runs are provided for you below. Format your output to match the sample output.

|  |  |  |
| --- | --- | --- |
| Enter x1: 4  Enter y1: 5  Enter x2: 7  Enter y2: 8  Distance: 4.242640687119285 | Enter x1: 0  Enter y1: 0  Enter x2: 3  Enter y2: 4  Distance: 5.0 | Enter x1: -2.7  Enter y1: 9  Enter x2: 0  Enter y2: -10.2  Distance: 19.38891435846783 |

**Problem 4 (IsFactor)**

* Write a program that has the class name IsFactor and that has the main method.
* The program should ask the user to enter two integers greater than 1.
* The program should determine whether the smaller of the two integers is a factor of the larger of the two integers (regardless of which order the two numbers are entered in).
* Based on whether either one is a factor of another, print out the appropriate message.
* As a reminder, a factor of one number is any whole number that can divide the number into evenly. For example, 8 can divide 64 evenly, therefore 8 is a factor of 64.
* Several sample runs are provided for you below. Format your output to match the sample output.

|  |
| --- |
| Enter the first number: 5  Enter the second number: 65  5 is a factor of 65 |

|  |
| --- |
| Enter the first number: 35  Enter the second number: 2  2 is not a factor of 35 |

**Problem 5 (RightAngledTriangle)**

* Write a program that has the class name RightAngledTriangle and that has the main method.
* The program should ask the user to enter three numbers a, b and c and check if they are the sides of right angled triangle.
* Using the Pythagorean theorem, you can check if it is a right-angled triangle or not. If it is a right-angled triangle, then print “Right angled triangle.”. If it is not, then print “Not a right-angled triangle. “.
* As per the theorem the sum of the square of two sides(legs) is equal to the square of the longest side(hypotenuse).
* Below is the formula given to calculate the hypotenuse and you must use Math.Sqrt and

Math.pow to calculate it. Check the style with prof.

h =

* Several sample runs are provided below. Format your output to match the sample output.

|  |
| --- |
| Enter a: 3  Enter b: 5  Enter c: 4  Right angled triangle |

|  |
| --- |
| Enter a: 9  Enter b: 5  Enter c: 12  Not a right angled triangle |

|  |
| --- |
| Enter a: 4.4  Enter b: 3.3  Enter c: 5.5  Right angled triangle |

**Problem 6 (EvenOrOdd)**

* Write a program that has the class name EvenOrOdd and that has the main method.
* The program should ask the user to enter a 4-bit binary number (1’s & 0’s) and then convert it into the decimal number.
* Assume that the user does not type any binary number that has leading zeros (For example: 0011).
* For example, if the input is 1101, the output should be 13 (Hint: Break the number into digits and then convert each digit to a value for a single digit).
* If the digits are n1, n2, n3 and n4, the decimal equivalent is 8n1 + 4n2 + 2n3 + 1n4.
* After you get the decimal value of the 4-bit number, print out the value and check if the value is even or odd and print accordingly.
* Several sample runs are provided for you below. Format your output to match the sample

output.

|  |
| --- |
| Enter a 4-bit integer: 1111  The decimal value of 0011 is 15  15 is an odd number |

|  |
| --- |
| Enter a 4-bit integer: 1010  The decimal value of 1010 is 10  10 is an even number |

**Problem 7 (CheckIfSame)**

* Write a program that has the class name CheckIfSame and that has the main method.
* The program should prompt the user to enter two integers.
* The program should determine whether the two integers have the same quotient when

divided by 10 and if they are divisible by 10.

* If the two integers have the same quotient, print out: "Quotients are the same!".
* If they are both divisible by 10, print out: "They are both divisible by 10!".
* If the two integers are both divisible by 10 and have the same quotient, then check whether the quotient is divisible by 2. If they are, print out: "Quotients are divisible by two!". If they are not, print out: "Quotients are not divisible by 2.".
* If the two integers are not divisible by 10, check whether the two integers have the same

remainder when divided by 10. If they do, print out: "Remainders are the same!".

* If they do not have the same quotient, are not divisible by 10, and do not have the same

remainder, then print "Nothing is the same!".

* Several sample runs are provided for you below. Format your output to match the sample

output. User input is indicated in bold.

|  |
| --- |
| First number: 10  Second number: 10  Quotients are the same!  They are both divisible by ten!  Quotients are not divisible by 2. |

|  |
| --- |
| First number: 3  Second number: 5  Quotients are the same! |

|  |
| --- |
| First number: 77  Second number: 98  Nothing is the same. |

|  |
| --- |
| First number: 120  Second number: 120  Quotients are the same!  They are both divisible by ten!  Quotients are divisible by two! |

|  |
| --- |
| First number: 15  Second number: 25  Remainders are the same! |

**Problem 8 (MaxMinMax)**

* Write a program that has the class name MaxMinMax and that has the main method.
* The program should prompt the user to enter two integers.
* The program should find the absolute value of each of the two integers and then check which number has the larger absolute value and display that value (match the sample output). If the two absolute values are equal, the program should print that they are equal.
* If the values are not the same, use the following equation to print a new maximum value (see output): MaxMin
* Several sample runs are provided for you below. Format your output to match the sample

output.

|  |
| --- |
| Enter n1: 2  Enter n2: 2  Values are equal. |

|  |
| --- |
| Enter n1: 2  Enter n2: 4  Absolute value of 4 is greater than 2  The new max is: 16 |

|  |
| --- |
| Enter n1: 10  Enter n2: 5  Absolute value of 10 is greater than 5  The new max is: 9765625 |