Christopher Bussen

CPS 150 02 – Algorithms and Programming 1

Lab 11

10/20/20

**Problem 1 Output**

The program will output the following:

Yes

No

**Problem 2 Outputs**

1. The program will output 1
2. The program will output 3
3. The program will output 5

**Problem 3 Code Segment**

if((m < 5) && (j < 0))

{

k = 3;

}

else if(m >= k)

{

k = 1;

}

**Problem 4 Algorithm**

1. Start the program
2. Import the scanner
3. Prompt the user to enter two boolean values
4. Declare two boolean variables to store the values from the user
5. Create a separate method for the or gate
6. Use an if/else statement and logical operators to determine if either value is true
7. Print a statement telling the user if either value is true or not
8. End or gate method
9. Create a separate method for the and gate
10. Use an if/else statement and logical operators to determine if both values are true
11. Print a statement telling the user if both values are true or not
12. End and gate method
13. Create a separate method for the not gate
14. Use an if/else statement and logical operators to determine if the two values are not true
15. Print a statement telling the user if the values are not true or not
16. End not gate method
17. Create a separate method for the xor gate
18. Use an if statement and logical operators to determine if exactly one of the values is true
19. Print a statement telling the user if exactly one value is true or not
20. End xor gate method
21. Create a separate method for the De Morgan’s Law
22. Use an if statement and logical operators to determine if both values are not true
23. Print a statement telling the user if both values aren’t true or not
24. End De Morgan’s Law method
25. Call all 5 methods in the main method
26. End main method
27. End the program

**Problem 4 Running Screenshot**

**Text

Description automatically generated**

**Problem 4 Code**

/\*

Christopher Bussen

CPS 150 02

Lab 11

BooleanOperators: boolean boolean; string string string string string

program takes in two boolean values from the user and uses separate methods to

determine if either of them are true, both are true, they're not both true,

exactly one of the two is true, and if neither are true

ex1: user inputs true, false - program outputs At least one true, Not both true, At least one true, One true, Not both true

ex2: user inputs true, true - program outputs At least one true, Both true, At least one true, More or less than 1 true, Both true

ex3: user inputs false, true - program outputs At least one true, Not both true, At least one true, One true, Not both true

ex4: user inputs false, false - program outputs Neither true, Not both true, Neither true, More or less than 1 true, Not both true

ex5: user inputs 0, 1 - program outputs error

ex6: user inputs alpha, omega - program outputs error

\*/

import java.util.Scanner;

public class BooleanOperators {

public static void main(String [] args){

//import scanner

Scanner input = new Scanner(System.in);

//prompt the user to enter two numbers and declare integer values for them

System.out.print("Please enter values for a and b (enter true or false): ");

boolean a = input.nextBoolean();

boolean b = input.nextBoolean();

//call all 5 operator methods

OrGate(a, b);

AndGate(a, b);

NotGate(a, b);

XorGate(a, b);

DeMorgan(a, b);

}

//create method for or gate

public static void OrGate(boolean x, boolean y){

//use if statement and logical operators to see if x or y is true

if(x || y){

System.out.println("At least one true");

}

else{

System.out.println("Neither true");

}

}

//create method for and gate

public static void AndGate(boolean x, boolean y){

//use if statement and logical operators to see if x and y are true

if(x && y){

System.out.println("Both true");

}

else{

System.out.println("Not both true");

}

}

//create method for not gate

public static void NotGate(boolean x, boolean y){

//use if statement and logical operators to see if x and y are not true

if(!x && !y){

System.out.println("Neither true");

}

else{

System.out.println("At least one true");

}

}

//create method for xor gate

public static void XorGate(boolean x, boolean y){

//use if statement and logical operators to see if exactly one of x and y is true

if((x && !y) || (!x && y)){

System.out.println("One true");

}

else{

System.out.println("More or less than 1 true");

}

}

//create method for De Morgan's Law

public static void DeMorgan(boolean x, boolean y){

//use if statement and logical operators to see if both x or y aren't true

if(!(x && y)){

System.out.println("Not both true");

}

else{

System.out.println("Both true");

}

}

}

**Problem 5 Algorithm**

1. Start the program
2. Import the scanner
3. Prompt the user to enter a 4 digit integer
4. Declare an int variable to store the variable of the 4 digit integer
5. Create separate method to do all of the iterations that will have an input of one integer - x (not necessary but prevents main method from getting cluttered)
6. Declare int variables for the remainder and the reverse number – they will be used in each iteration
7. Find the remainder of dividing the method’s input by 10 (%10) and store the value in the remainder variable
8. Reassign the reverse number variable to be the current value of the reverse number \* 10 (since a digit is being added) then add the remainder
9. Divide x (function input) by 10
10. Repeat steps 7 – 10 three more times, using each variable’s new value the next time through
11. Print the reverse number of the original 4 digit integer
12. End the iteration method
13. Call the iteration method in the main method using the 4 digit integer as the input
14. End the main method
15. End the program

**Problem 5 Running Screenshot**

**Graphical user interface, text

Description automatically generated**

**Problem 5 Code**

/\*

Christopher Bussen

CPS 150 02

Lab 11

ReverseOf4Numbers: number ; number

program takes in an 4 digit integer value from the user and prints out

the value in the reverse order

ex1: user inputs 3167 - program outputs 7613

ex2: user inputs -1054 - program outputs -4501

ex3: user inputs 101.65 - program outputs error

ex4: user inputs x - program outputs error

ex5: user inputs 10189 - program outputs 9810 (will only print reverse of 4 numbers)

ex6: user inputs 78 - program outputs 8700 (4 iterations will happen - 0 will be printed for missing numbers)

\*/

import java.util.Scanner;

public class ReverseOf4Numbers {

public static void main(String [] args){

//import scanner

Scanner input = new Scanner(System.in);

//prompt user to input the number they want to be reversed and declare an int variable to store the value

System.out.print("Enter the 4-digit integer you want reversed: ");

int number = input.nextInt();

//call ReverseIteration method

ReverseIteration(number);

}

public static void ReverseIteration(int x){

//declare int variables for the remainder and reverse number - will be used in iterations

int remainder = 0;

int reverseNumber = 0;

//find remainder of dividing by 10

//multiply the reverse number by 10 since were adding another digit then add the remainder

//divide the number by 10 and use that as the new number

//repeat process 3 more times

remainder = x % 10;

reverseNumber = reverseNumber \* 10 + remainder;

x = x / 10;

remainder = x % 10;

reverseNumber = reverseNumber \* 10 + remainder;

x = x / 10;

remainder = x % 10;

reverseNumber = reverseNumber \* 10 + remainder;

x = x / 10;

remainder = x % 10;

reverseNumber = reverseNumber \* 10 + remainder;

x = x / 10;

//print the reversed number

System.out.println("Your number reversed is " + reverseNumber);

}

}