Common Programming Problems

Java Data Types

from: https://www.w3resource.com/java-exercises/datatypes/index.php

Java Conditional Statements

from: https://www.w3resource.com/java-exercises/conditional-statement/index.php

Java Methods

from: https://www.w3resource.com/java-exercises/method/index.php

1. to break an integer into a sequence of individual digits

```
import java.util.Scanner;

public class jdt_10 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Input six non-negtive digits: ");
        int num = in.nextInt();
        int[] numArr = new int[6];
        numArr[0] = (num/100000)%10;
        numArr[1] = (num/10000)%10;
        numArr[2] = (num/1000)%10;
        numArr[3] = (num/100)%10;
        numArr[4] = (num/10)%10;
        numArr[5] = (num%10);
        //System.out.println(Arrays.toString(numArr));
        for (int i = 0; i < 6; i++){
            System.out.print(numArr[i] + " ");
        }
        in.close();
    }
}
```

2. to solve quadratic equations (use if, else if and else)

```
import java.util.Scanner;

public class jcs_02 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Input a: ");
        double a = in.nextDouble();
        System.out.print("Input b: ");
        double b = in.nextDouble();
        System.out.print("Input c: ");
        double c = in.nextDouble();

        double deter = b * b - 4.0 * a * c;

        if (deter > 0){
            double root1 = (-b + Math.pow(deter, 0.5)) / (2.0 * a);
            double root2 = (-b - Math.pow(deter, 0.5)) / (2.0 * a);
            System.out.println("The roots are " + root1 + " and " + root2);
        }
        else if (deter == 0){
            double root = -b / (2.0 * a);
            System.out.println("The root is " + root);
        }
        else{
            System.out.println("The equation has no real roots");
        }

        in.close();
    }
}
```

3. program that keeps a number from the user and generates an integer between 1 and 7 and displays the name of the weekday

```
import java.util.Scanner;
public class jcs_05 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Input number: ");
        int day = in.nextInt();

        System.out.println(getDayName(day));

        in.close();
    }

    public static String getDayName(int day) {
        String dayName = "";
        switch (day) {
            case 1: dayName = "Sunday"; break;
            case 2: dayName = "Monday"; break;
            case 3: dayName = "Tuesday"; break;
            case 4: dayName = "Wednesday"; break;
            case 5: dayName = "Thursday"; break;
            case 6: dayName = "Friday"; break;
            case 7: dayName = "Saturday"; break;
            default:dayName = "Invalid day range";
        }

        return dayName;
    }
}
```

4. program that reads in two floating-point numbers and tests whether they are the same up to three decimal places

```
import java.util.Scanner;
public class jcs_06 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Input floating-point number: ");
        double x = in.nextDouble();
        System.out.print("Input floating-point another number: ");
        double y = in.nextDouble();

        x = Math.round(x * 1000);
        x = x / 1000;

        y = Math.round(y * 1000);
        y = y / 1000;

        if (x == y){
            System.out.println("They are the same up to three decimal places");
        }
        else{
            System.out.println("They are different");
        }

        in.close();
    }
}
```

5. program that takes a year from user and print whether that year is a leap year or not.

```
import java.util.Scanner;
public class jcs_09 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        System.out.print("Input the year: ");
        int year = in.nextInt();

        boolean con1 = (year % 4) == 0;
        boolean con2 = (year % 100) != 0;
        boolean con3 = ((year % 100 == 0) && (year % 400 == 0));

        if (con1 && (con2 || con3)){
            System.out.println(year + " is a leap year");
        }
        else{
            System.out.println(year + " is not a leap year");
        }

        in.close();
    }
}
```

6. to display the multiplication table of a given integer

```
import java.util.Scanner;
public class jcs_14 {
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        System.out.print("Input the number(Table to be calculated): ");
        System.out.print("Input number of terms : ");
      int n = in.nextInt();
        for(int i = 0; i <= n; i++){
            System.out.println(n + " X " + i + " = " + n*i);
        }

        in.close();
    }
}
/*
Output:
Input the number(Table to be calculated): Input number of terms : 4
4 X 0 = 0
4 X 1 = 4
4 X 2 = 8
4 X 3 = 12
4 X 4 = 16
*/
```

7. to display the pattern like right angle triangle with a number

```
import java.util.Scanner;
public class jcs_16 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Input number of rows : ");
        int n = in.nextInt();
        for (int i = 1; i <= n; i++){
            for (int j = 1; j <= i; j++){
                System.out.print(j);
            }
            System.out.println("");
        }

        in.close();
    }
}
/*
Output:
Input number of rows : 4
1
12
123
1234
*/
```

8. to make such a pattern like right angle triangle with a number which will repeat a number in a row

```
import java.util.Scanner;
public class jcs_17 {
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        System.out.print("Input number of n : ");
        int n = in.nextInt();
        for (int i = 1; i <=n ; i++){
          for (int j = 1; j <= i; j++){
              System.out.print(i);
            }
            System.out.println("");
        }
        in.close();
    }
}
/*
Output:
Input number of n : 4
1
22
333
4444
*/
```

9. to make such a pattern like right angle triangle with number increased by 1

```
import java.util.Scanner;
public class jcs_18 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int k = 1;
        System.out.print("Input number of rows : ");
        int n = in.nextInt();
        for (int i = 1; i <= n; i++){
             for (int j = 1; j <= i; j++){
                  System.out.print(k++ + " ");
             }
             System.out.println("");
        }

        in.close();
    }
}
/*
Output:
Input number of rows : 4
1
2 3
4 5 6
7 8 9 10
*/
```

10. to make such a pattern like a pyramid with a number which will repeat the number in the same row

```
import java.util.Scanner;
public class jcs_19 {
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        System.out.print ("Input number of rows : ");
        int n = in.nextInt();
        int s = n + 4-1;
        for(int i = 1; i <= n; i++){
            for(int x = s; x != 0; x--){
                System.out.print(" ");
            }
            for(int j = 1; j <= i; j++){
                System.out.print(i + " ");
            }
            System.out.println();
            s--;
        }

        in.close();
    }
}
/*
Output:
Input number of rows : 4
       1
      2 2
     3 3 3
    4 4 4 4
*/
```

11. program in Java to print the Floyd's Triangle

```
import java.util.Scanner;
public class jcs_20 {
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        System.out.print("Input number of rows : ");
    int numberOfRows = in.nextInt();
        int number = 1;
        for (int row = 1; row <= numberOfRows; row++){
            for (int column = 1; column <= row; column++){
                System.out.print(number + " ");
                number++;
            }
            System.out.println();
        }

        in.close();
    }
}
/*
Output:
Input number of rows : 4
1
2 3
4 5 6
7 8 9 10
*/
```

12. to display the pattern like a diamond

```
import java.util.Scanner;
public class jcs_21 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Input number of rows (half of the diamond) : ");
        int r = in.nextInt();
        for(int i = 0; i <= r; i++){
            for(int j = 1; j <= r-i; j++){
                System.out.print(" ");
            }
            for(int j = 1; j <= 2*i-1; j++){
                System.out.print("*");
            }
            System.out.print("\n");
        }

        for(int i = r-1; i >= 1; i--){
            for(int j = 1; j <= r-i; j++){
                System.out.print(" ");
            }
            for(int j = 1; j <= 2*i-1; j++){
                System.out.print("*");
            }
            System.out.print("\n");
        }

        in.close();
    }
}
/*
Output:
Input number of rows (half of the diamond) : 4

   *
  ***
 *****
*******
 *****
  ***
   *
*/
```

13. to display Pascal's triangle

```
import java.util.Scanner;
public class jcs_22 {
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        int c = 1;
        System.out.print("Input number of rows: ");
        int no_row = in.nextInt();
        for(int i = 0; i < no_row; i++){
            for(int blk = 1; blk <= no_row-i; blk++){
                System.out.print(" ");
            }
            for(int j = 0; j <= i; j++){
                if (j==0||i==0)
                    c=1;
                else
                    c=c*(i-j+1)/j;
                System.out.print(" "+c);
            }
            System.out.print("\n");
        }

        in.close();
    }
}
/*
Output:
Input number of rows: 4
     1
    1 1
   1 2 1
  1 3 3 1
*/
```

14. to generate a following *'s triangle

```
import java.util.Scanner;
public class jcs_23 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Input the number: ");
        int n = in.nextInt();
        for (int i = n; i > 0; i--) {
            for (int spc = n - i; spc > 0; spc--){
                System.out.print(" ");
            }
            for (int j = 0; j < i; j++){
                System.out.print("*");
            }
            System.out.println();
        }

        in.close();
    }
}
/*
Output:
Input the number: 4
****
 ***
  **
   *
*/
```

15. to generate a following @'s triangle

```
import java.util.Scanner;
public class jcs_24 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Input the number: ");
        int n = in.nextInt();

        for (int i = 0; i < n; i++){
            for (int spc = n - i; spc > 0; spc--){
                System.out.print(" ");
            }
            for (int j = 0; j <= i; j++){
                System.out.print("@");
            }
            System.out.println();
        }

        in.close();
    }
}
/*
Output:
Input the number: 4
    @
```

```
   @@
  @@@
 @@@@
*/
```

16. to display the number rhombus structure

```
import java.util.Scanner;
public class jcs_25 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Input the number:  ");
        int n = in.nextInt();
        int count = 1;
        int no_of_spaces = 1;
        int start = 0;

        for (int i = 1; i < (n * 2); i++){
            for (int spc = n - no_of_spaces; spc > 0; spc--){
                System.out.print(" ");
            }
            if (i < n){
                start = i;          //for number
                no_of_spaces++;    //for spaces
            }
            else{
                start = n * 2 - i;   //for number
                no_of_spaces--;      //for space
            }
            for (int j = 0; j < count; j++){
                System.out.print(start);
                if (j < count / 2){
                    start--;
                }
                else{
                    start++;
                }
            }
            if (i < n){
                count = count + 2;
            }
            else {
                count = count - 2;
            }
            System.out.println();
        }

        in.close();
    }
}
/*
Output:
Input the number:  4
   1
  212
 32123
4321234
 32123
  212
   1
*/
```

17. to display the following character rhombus structure

```
import java.util.Scanner;
public class jcs_26 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Input the number: ");
        int n = sc.nextInt();
        int count = 1;
        int count2 = 1;
        char c = 'A';

        for (int i = 1; i < (n * 2); i++){
            for (int spc = n - count2; spc > 0; spc--){
                System.out.print(" ");
            }
            if (i < n){
                count2++;
            }
            else{
                count2--;
            }
            for (int j = 0; j < count; j++){
                System.out.print(c);//print Character
                if (j < count / 2){
                    c++;
                }
                else{
                    c--;
                }
            }
            if (i < n){
                count = count + 2;
            }
            else{
                count = count - 2;
            }
            c = 'A';

            System.out.println();
        }

        sc.close();
    }
}
/*
Input the number: 4
   A
  ABA
 ABCBA
ABCDCBA
 ABCBA
  ABA
   A
*/
```

18. program that reads an positive integer and count the number of digits the number (less than ten billion) has

```
import java.util.Scanner;
public class jcs_29 {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Input an integer number less than ten billion: ");

        if (in.hasNextLong()){
            long n = in.nextLong();

            if (n < 0){
                n *= -1;
            }
            if (n >= 10000000000L){
                System.out.println("Number is greater or equals 10,000,000,000!");
            }
            else{
                int digits = 1;
                if (n >= 10 && n < 100){
                    digits = 2;
                }
                else if (n >= 100 && n < 1000){
                    digits = 3;
                }
                else if (n >= 1000 && n < 10000){
                    digits = 4;
                }
                else if (n >= 10000 && n < 100000){
                    digits = 5;
                }
                else if (n >= 100000 && n < 1000000){
                    digits = 6;
                }
                else if (n >= 1000000 && n < 10000000){
                    digits = 7;
                }
                else if (n >= 10000000 && n < 100000000){
                    digits = 8;
                }
                else if (n >= 100000000 && n < 1000000000){
                    digits = 9;
                }
                else if (n >= 1000000000 && n < 10000000000L){
                    digits = 10;
                }
                System.out.println("Number of digits in the number: " + digits);
            }
        }
        else{
            System.out.println("The number is not an integer");
        }

        in.close();
    }
}
```

19. program that accepts two floating point numbers and checks whether they are the same up to two decimal places

```
import java.util.Scanner;
public class jcs_32 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Input first floating point number: ");
        double num1 = input.nextDouble();
        System.out.print("Input second floating point number: ");
        double num2 = input.nextDouble();
        input.close();

        if (Math.abs(num1 - num2) <= 0.01) {
            System.out.println("These numbers are the same.");
        }
        else {
            System.out.println("These numbers are different.");
        }
    }
}
```

20. to display the middle character of a string. Note:

a) If the length of the string is odd there will be two middle characters.

b) If the length of the string is even there will be one middle character.

`point to be noted:`

```
String str = System.console().readLine(); //does exactly the same as

Scanner scan = new Scanner(System.in);
String str = scan.nextLine();
```

```
public class jme_03 {
    public static void main(String[] args) {
        System.out.print("Input a string: ");
        String str = System.console().readLine(); //takes input from console, also can be done by creating an object of Scanner class
        System.out.println("The string is: " + str);
        if (str.length()%2 == 0){
            System.out.println("The middle two characters in the string: " + str.substring(str.length()/2-1, str.length()/2+1));
        } else {
            System.out.println("The middle character in the string: " + str.charAt(str.length()/2));
        }
    }
}
```

21. to count all vowels in a string

```
public class jme_04 {
    public static void main(String[] args) {
        System.out.print("Input a string: ");
        String str = System.console().readLine();
        vowelCount(str);
    }
    public static void vowelCount(String str){
        int count = 0;
        for (int i = 0; i < str.length(); i++){
            if (str.charAt(i) == 'a' || str.charAt(i) == 'e' || str.charAt(i) == 'i' || str.charAt(i) == 'o' || str.charAt(i) == 'u' |
            str.charAt(i) == 'A' || str.charAt(i) == 'E' || str.charAt(i) == 'I' || str.charAt(i) == 'O' || str.charAt(i) == 'U'){
                count++;
            }
        }
        System.out.println("Number of vowels in the string: " + count);
    }
}
```

22. to count all words in a string

```
public class jme_05 {
    public static void main(String[] args) {
        System.out.print("Input a string: ");
        String str = System.console().readLine();
        wordCount(str);
    }
    public static void wordCount(String str){
        int count = 0;
        for (int i = 0; i < str.length(); i++){
            if (str.charAt(i) == ' '){
                count++;
            }
        }
        System.out.println("Number of words in the string: " + (count+1));
    }
}
```

23. to compute the sum of the digits in an integer

point to be noted:

```
int num = Integer.parseInt(System.console().readLine()); //does exactly same as

Scanner scan = new Scanner(System.in);
int num = scan.nextInt();
```

```
public class jme_06 {
    public static void main(String[] args) {
        System.out.print("Input the number: ");
        int num = Integer.parseInt(System.console().readLine());
        sumOfDigits(num);
    }
    public static void sumOfDigits(int num){
        int sum = 0;
        while (num > 0){
            sum += num%10;
            num /= 10;
        }
        System.out.println("The sum of digits is: " + sum);
    }
}
```

24. to display the first n pentagonal numbers. Note:

A pentagonal number is a figurate number that extends the concept of triangular and square numbers to the pentagon, but, unlike the first two, the patterns involved in the construction of pentagonal numbers are not rotationally symmetrical.

```
public class jme_07 {
    public static void main(String[] args) {
        System.out.print("The first pentagonal numbers till: ");
        int n = Integer.parseInt(System.console().readLine());
        pentagonalNumber(n);
    }
    public static void pentagonalNumber(int n){
        int pentagonal = 0;
        for (int i = 1; i <= n; i++){
            pentagonal = i*(3*i-1)/2;
            //Pn = (n-1)n + n(n+1)/2
            System.out.printf("%-6d",pentagonal);
            if (i%10 == 0){
                System.out.println();
            }
        }
    }
}
```

25. to compute the future investment value at a given interest rate for a specified number of years

```
public class jme_08 {
    public static void main(String[] args) {
        System.out.print("The amount invested: ");
        double investmentAmount = Double.parseDouble(System.console().readLine());
        System.out.print("Annual interest rate: ");
        double annualInterestRate = Double.parseDouble(System.console().readLine());
        System.out.print("Number of years: ");
        int years = Integer.parseInt(System.console().readLine());
        double monthlyInterestRate = annualInterestRate/1200;

        futureInvestmentValue(investmentAmount, monthlyInterestRate, years);
    }
    public static void futureInvestmentValue(double investmentAmount, double monthlyInterestRate, int years){
        double futureInvestmentValue = 0;
        System.out.printf("%5s" + "%13s%n","Years","Future Value");
        for (int i = 1; i <= years; i++){
            futureInvestmentValue = investmentAmount * Math.pow((1 + monthlyInterestRate *12 ), i);
            System.out.printf("%5d" + " %11.2f%n",i, futureInvestmentValue);
        }
    }
}
```

26. to print characters between two characters (i.e. A to P)

```
public class jme_09 {
    public static void main(String[] args) {
        System.out.print("Enter the first character: ");
        char ch1 = System.console().readLine().charAt(0);
        System.out.print("Enter the second character: ");
        char ch2 = System.console().readLine().charAt(0);
        System.out.print("Enter the number of characters per line: ");
        int numberPerLine = Integer.parseInt(System.console().readLine());
        printChars(ch1, ch2, numberPerLine);
    }

    public static void printChars(char ch1, char ch2, int numberPerLine){
        int count = 0;
        for (int i = ch1; i <= ch2; i++){
            System.out.print((char)i + " ");
            count++;
            if (count % numberPerLine == 0){
                System.out.println();
            }
        }
    }
}
```

27. to check whether a string is a valid password

Password rules:

A password must have at least ten characters.

A password consists of only letters and digits.

A password must contain at least two digits.

```
public class jme_11 {
    public static void main(String[] args) {
        System.out.print("1. A password must have at least eight characters.\n" +
                         "2. A password consists of only letters and digits.\n" +
                         "3. A password must contain at least two digits.\n");
        System.out.print("Enter a password: ");
        String password = System.console().readLine();
        checkValidPassword(password);
    }
    public static void checkValidPassword(String password){
        int count = 0;
        if (password.length() < 8){
            System.out.println("Invalid password.");
            return;
        }
        for (int i = 0; i < password.length(); i++){
            if (Character.isLetterOrDigit(password.charAt(i))){
                if (Character.isDigit(password.charAt(i))){
                    count++;
                }
            } else {
                System.out.println("Invalid password.");
                return;
            }
        }
        if (count < 2){
            System.out.println("Invalid password.");
            return;
        }
        System.out.println("Valid password.");
    }
}
```

28. to displays an n-by-n matrix

```
public class jme_12 {
    public static void main(String[] args) {
        System.out.print("Enter a number: ");
        int n = Integer.parseInt(System.console().readLine());
        nByNMatrix(n);
    }

    public static void nByNMatrix(int n){
        int[][] matrix = new int[n][n];
        for (int i = 0; i < n; i++){
            for (int j = 0; j < n; j++){
                matrix[i][j] = (int)(Math.random() * 2);
            }
        }
        for (int i = 0; i < n; i++){
            for (int j = 0; j < n; j++){
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

29. to find all twin prime numbers less than n

```
//Write a Java method to find all twin prime numbers less than n
public class jme_16 {
    public static void main(String[] args) {
        System.out.print("Twin prime numbers till: ");
        int n = Integer.parseInt(System.console().readLine());

        for (int i = 2; i < n; i++) {
            if (checkPrime(i) && checkPrime(i+2)){
                System.out.println("(" + i + ", " + (i+2) + ")");
            }
        }
    }

    public static boolean checkPrime(int n){
        for (int i = 2; i < n; i++) {
            if (n % i == 0){
                return false;
            }
        }
        return true;
    }
}
```

30. to count the number of digits in an integer that have the value 2. The integer may be assumed to be non-negative.

```
public class jme_17 {
    public static void main(String[] args) {
        System.out.print("Enter a number: ");
        int n = Integer.parseInt(System.console().readLine());
        if (n>0){
            twoCount(n);
        }

    }

    public static void twoCount(int n){
        int count = 0;
        do{
            if (n % 10 == 2){
                count++;
            }
            n /= 10;
        } while(n>0);
        System.out.println("Number of 2's: " + count);
    }
}
```

31. to develop a Java method for extracting the first digit from a positive or negative integer

```
public class jme_20 {
    public static void main(String[] args) {
        System.out.print("Input an integer: ");
        int num = Integer.parseInt(System.console().readLine());
        extractingFirstDigit(num);
        System.out.println("The first digit of the number is: " + extractingFirstDigit(num));
    }

    public static int extractingFirstDigit(int num){
        int factor = 10;
        while (num/factor != 0){
            factor *= 10;
        }
        int firstDigit = Math.abs(num / (factor/10));
        return firstDigit;
    }
}
```

32. to display the factors of 3 in a given integer

```
public class jme_21 {
    public static void main(String[] args) {
        System.out.print("Input an integer(positive/negative): ");
        int num = Integer.parseInt(System.console().readLine());
        System.out.print("\nFactors of 3 in the number are: ");
        checkFactor(num);
    }

    public static void checkFactor(int num){
        System.out.print(num + " = ");
        int result = num;
        while (result%3 == 0){
            System.out.print("3 * ");
            result /= 3;
        }
        System.out.println(result);
    }
}
```