Index

[Exercise 1: OddEvenSum 4](#_Toc65790327)

[Task 1 - Create a OddEvenSum Classs 4](#_Toc65790328)

[Task 2 - Write code for OddEvenSum Class 4](#_Toc65790329)

[Task 3 - Execute your program 5](#_Toc65790330)

[Exercise 2: Compute PI 5](#_Toc65790331)

[Task 1 - Create a ComputePI Classs 5](#_Toc65790332)

[Task 2 - Write code for ComputePI Class 5](#_Toc65790333)

[Task 3 - Execute your program 6](#_Toc65790334)

[Exercise 3: Do ... while statement 6](#_Toc65790335)

[Task 1 - Create a DoWhile Classs 6](#_Toc65790336)

[Task 2 - Write code for DoWhile Class 6](#_Toc65790337)

[Task 3 - Execute your program 7](#_Toc65790338)

[Exercise 4: Array 7](#_Toc65790339)

[Task 1 - Create a ArrayTest class 7](#_Toc65790340)

[Task 2 - Write your code inside the main() method 7](#_Toc65790341)

[Task 3 - Execute your program 8](#_Toc65790342)

[Exercise 5: Loops and conditional statement 8](#_Toc65790343)

[Task 1 - Create a PrimeList class 8](#_Toc65790344)

[Task 2 - Write your code inside the main() method 9](#_Toc65790345)

[Task 3 - Execute your program 10](#_Toc65790346)

[Exercise 6: Nested loop 10](#_Toc65790347)

[Task 1 - Create a PrintPattern class 10](#_Toc65790348)

[Task 2 - Write code inside the main() method 11](#_Toc65790349)

[Task 3 - Execute your program 12](#_Toc65790350)

[Exercise 7: Break label 12](#_Toc65790351)

[Task 1 - Create a JavaGoto class 12](#_Toc65790352)

[Task 2 - Write code for JavaGoto class 12](#_Toc65790353)

[Task 3 - Execute your program 13](#_Toc65790354)

[Exercise 8: Continue label 13](#_Toc65790355)

[Task 1 - Create a JavaGoto class 13](#_Toc65790356)

[Task 2 - Write code for ContinueLabel class 13](#_Toc65790357)

[Task 3 - Execute your program 14](#_Toc65790358)

[Exercise 9: Array Sort 14](#_Toc65790359)

[Task 1 - Create a ArrayBubleSort class 14](#_Toc65790360)

[Task 2 - Write code for ArrayBubleSort class 14](#_Toc65790361)

[Task 3 - Execute your program 15](#_Toc65790362)

[Exercise 10: Reverse String 15](#_Toc65790363)

[Task 1 - Create a ReverseString class 15](#_Toc65790364)

[Task 2 - Write code for ReverseString class 15](#_Toc65790365)

[Task 3 - Execute your program 16](#_Toc65790366)

[Exercise 11: Array of String 16](#_Toc65790367)

[Task 1 - Create a Dayofweek class 16](#_Toc65790368)

[Task 2 - Write code for Dayofweek class 17](#_Toc65790369)

[Task 3 - Execute your program 17](#_Toc65790370)

[Exercise 12: Nested Loop 17](#_Toc65790371)

[Exercise 13: Nested Loop 17](#_Toc65790372)

[Exercise 14: CozaLozaWoza 18](#_Toc65790373)

[Exercise 15: Fibonacci 18](#_Toc65790374)

[Exercise 16: Special Characters and Escape Sequences 19](#_Toc65790375)

[Exercise 17: Print Pattern using nested-loop 19](#_Toc65790376)

[Exercise 18: Print Number using nested-loop 20](#_Toc65790377)

[Exercise 19: Pascal Triangle 20](#_Toc65790378)

[Exercise 20: Random array 21](#_Toc65790379)

[Exercise 21: GCD Program 21](#_Toc65790380)

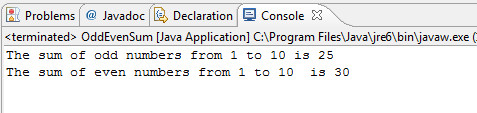
**Next Step**

OddEvenSum

## Write code for OddEvenSum Class

* 1. **public** **class** OddEvenSum {
  2. **public** **static** **void** main(String[] args) {
  3. **int** lowerbound = 1;
  4. **int** upperbound = 10;
  5. **int** sumOdd = 0;
  6. **int** sumEven = 0;
  7. **for** (**int** number = lowerbound; number <= upperbound; number++) {
  8. **if** (number % 2 == 0) {
  9. sumEven += number;
  10. } **else** {
  11. sumOdd += number;
  12. }
  13. }
  14. // Print the result
  15. System.*out*.println("The sum of odd numbers from " + lowerbound + " to "
  16. + upperbound + " is " + sumOdd);
  17. System.*out*.println("The sum of even numbers from " + lowerbound
  18. + " to " + upperbound + "is " + sumEven);
  19. }
  20. }

## Execute your program



1. Compute PI

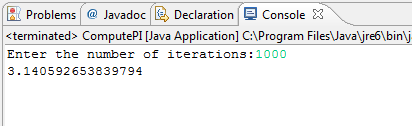
Write a program called ComputePI to compute the value of π, using the following series expansion. You have to decide on the termination criterion used in the computation (such as the number of terms used or the magnitude of an additional term). Is this series suitable for computing π?

http://www3.ntu.edu.sg/home/ehchua/programming/java/images/ExerciseBasics_ComputePI.png

## Create a ComputePI Classs

* 1. **public** **class** ComputePI {
  2. **public** **static** **void** main(String[] args) {
  3. **double** pi = 0;
  4. **double** i = 0;
  5. **int** sign = 1;
  6. Scanner scanner = **new** Scanner(System.*in*);
  7. System.*out*.print("Enter the number of iterations:");
  8. **int** Number\_Of\_Iterations = scanner.nextInt();
  9. **while** (i < Number\_Of\_Iterations) {
  10. pi = pi + 4 \* sign / (2 \* i + 1);
  11. i++;
  12. sign = -sign;
  13. }
  14. System.*out*.println(pi);
  15. }
  16. }

## Execute your program

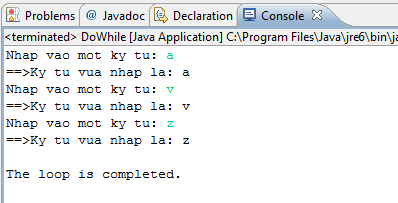


Do ... while statement

## Write code for DoWhile Class

* 1. **public** **class** DoWhile {
  2. **public** **static** **void** main(String[] args) {
  3. **char** ch;
  4. Scanner scanner = **new** Scanner(System.*in*);
  5. **do**{
  6. System.*out*.print("Nhap vao mot ky tu: ");
  7. ch = (scanner.next()).charAt(0);
  8. System.*out*.printf("==>Ky tu vua nhap la: %c\n", ch);
  9. }**while**(ch!='z');
  10. System.*out*.print("\nThe loop is completed. \n");
  11. }
  12. }

## Execute your program



1. Array

## Create a ArrayTest class

### Declare a array and initiate the values for this array

* 1. **int**[] marks = {74, 43, 58, 60, 90, 64, 70};

### Declare variables to contain sum and average of the array

* 1. **int** sum = 0;
  2. **double** avg;

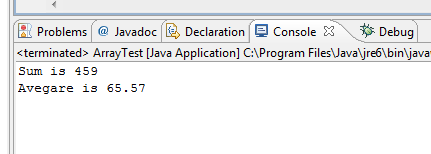
### Use for loop to calculate sum and average of the array

* 1. **int** count = marks.length;
  2. **for** (**int** i=0; i<count; i++) {
  3. sum += marks[i];
  4. }
  5. avg = (**double**)sum/count;

### Print the output to console window

* 1. System.*out*.printf("Sum is %d\n", sum);
  2. System.*out*.printf("Avegare is %.2f\n", avg);

## Execute your program

* 1. 

Loops and conditional statement

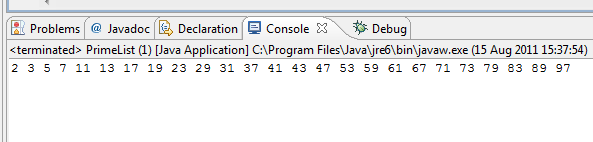
Write a loop to print all prime number between 2 and 100

## Create a PrimeList class

Enter the following codes to find all prime numbers:

* 1. **int** upperbound = 100;
  2. **for** (**int** number = 2; number <= upperbound; number++) {
  3. // Not prime, if there is a factor between 2 and sqrt of number
  4. **int** maxFactor = (**int**)Math.*sqrt*(number);
  5. **boolean** isPrime = **true**;
  6. **int** factor = 2;
  7. **while** (isPrime && factor <= maxFactor) {
  8. **if** (number % factor == 0) { // Factor of number?
  9. isPrime = **false**;
  10. }
  11. factor++;
  12. }
  13. **if** (isPrime)
  14. System.*out*.print(number + " ");
  15. }

## Execute your program

* 1. 

Nested loop

In this exercise, you will use a **nested loop** to print this pattern:

########

#######

######

#####

####

###

##

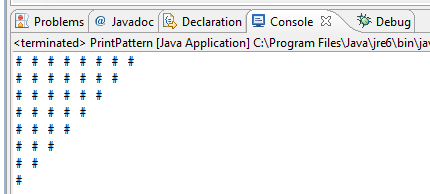
#

## Create a PrintPattern class

Enter the following codes to print this pattern

* 1. **int** size = 8;
  2. **for** (**int** row = 1; row <= size; row++) {
  3. **for** (**int** col = row; col <= size; col++) {
  4. System.*out*.print("# ");
  5. }
  6. System.*out*.println();
  7. }

## Execute your program



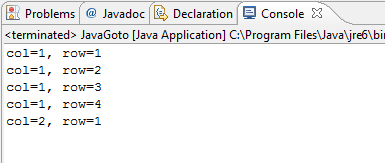
Break label

This program will action like goto statement in C

## Create a JavaGoto class

* 1. **public** **class** JavaGoto {
  2. **public** **static** **void** main(String[] args) {
  3. **int** max = 4;
  4. **int** limit = 2;
  5. out:
  6. **for** (**int** row = 1; row <= max; row++) {
  7. **for** (**int** col = 1; col <= max; col++) {
  8. System.*out*.printf("col=%d, row=%d\n", row, col);
  9. **if** (row == limit) {
  10. **break** out;
  11. }
  12. }
  14. }
  15. }
  16. }

## Execute your program

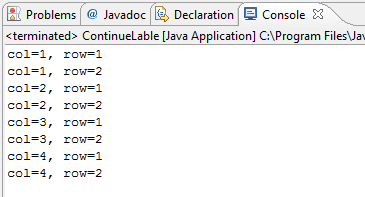
* 1. 

Continue label

## Write code for ContinueLabel class

* 1. **public** **class** ContinueLable {
  2. **public** **static** **void** main(String[] args) {
  3. **int** max = 4;
  4. **int** cancel = 2;
  5. out: **for** (**int** row = 1; row <= max; row++) {
  6. **for** (**int** col = 1; col <= max; col++) {
  7. System.*out*.printf("col=%d, row=%d\n", row, col);
  8. **if** (col == cancel) {
  9. **continue** out;
  10. }
  11. }
  12. }
  13. }
  14. }

## Execute your program

* 1. 

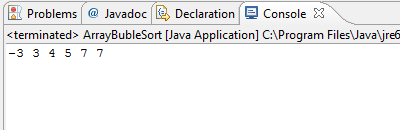
Array Sort

This program will use bubble sort algorithm

## Write code for ArrayBubleSort class

* 1. **public** **class** ArrayBubleSort {
  2. **public** **static** **void** main(String[] args) {
  3. **int** temp;
  4. **int**[] a = { 4, 5, 3, 7, -3, 9, 7 };
  5. **for** (**int** i = 0; i < a.length - 1; i++) {
  6. **for** (**int** j = i; j <= a.length - 1; j++) {
  7. **if** (a[i] > a[j]) {
  8. temp = a[i];
  9. a[i] = a[j];
  10. a[j] = temp;
  11. }
  12. }
  13. }
  14. **for** (**int** i = 0; i < a.length - 1; i++) {
  15. System.*out*.print(a[i] + " ");
  16. }
  17. }
  18. }

## Execute your program



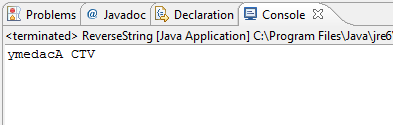
Reverse String

Write a program to reverse a string. It invokes the String method charAt(i), which returns the ith character in the string, counting from 0.

## Write code for ReverseString class

* 1. **public** **class** ReverseString {
  2. **public** **static** **void** main(String[] args) {
  3. String palindrome = "VTC Academy";
  4. **int** len = palindrome.length();
  5. **char**[] tempCharArray = **new** **char**[len];
  6. **char**[] charArray = **new** **char**[len];
  8. // put original string in an array of chars
  9. **for** (**int** i = 0; i < len; i++) {
  10. tempCharArray[i] = palindrome.charAt(i);
  11. }
  13. // reverse array of chars
  14. **for** (**int** j = 0; j < len; j++) {
  15. charArray[j] = tempCharArray[len - 1 - j];
  16. }
  18. String reversePalindrome = **new** String(charArray);
  19. System.*out*.println(reversePalindrome);
  20. }
  21. }

## Execute your program

* 1. 

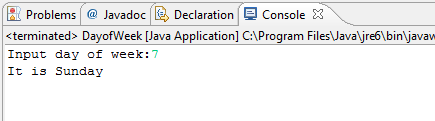
1. Array of String

Write a program to take a number and return the day of the week

## Write code for Dayofweek class

* 1. **public** **class** DayofWeek {
  2. **public** **static** **void** main(String[] args) {
  3. Scanner scanner = **new** Scanner(System.*in*);
  4. String[] calendarDays = { "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday" };
  5. System.*out*.print("Input day of week:");
  6. **int** dayofweek = scanner.nextInt();
  7. System.*out*.println("It is " + calendarDays[dayofweek - 1]);
  8. }
  9. }

## Execute your program



1. Nested Loop

Write a program to display this number to console screen:

1 2 3 4 5 6 7 8 9 10

2 4 6 8 10 12 14 16 18 20

3 6 9 12 15 18 21 24 27 30

4 8 12 16 20 24 28 32 36 40

5 10 15 20 25 30 35 40 45 50

6 12 18 24 30 36 42 48 54 60

7 14 21 28 35 42 49 56 63 70

8 16 24 32 40 48 56 64 72 80

9 18 27 36 45 54 63 72 81 90

10 20 30 40 50 60 70 80 90 100

1. Nested Loop

Write a program that draws the following figures:

\* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \* \*

1. CozaLozaWoza

Write a program called **CozaLozaWoza** which prints the number 1 to 110, 11 numbers per line. The program shall print "Coza" in place of the numbers which are multiples of 3, "Loza" for multiples of 5, "Woza" for multiples of 7, "CozaLoza" for multiples of 3 and 5, and so on. The output shall look like:

* 1. 1 2 Coza 4 Loza Coza Woza 8 Coza Loza 11
  2. Coza 13 Woza CozaLoza 16 17 Coza 19 Loza CozaWoza 22
  3. 23 Coza Loza 26 Coza Woza 29 CozaLoza 31 32 Coza
  4. ......

Hints:

* 1. **public** **class** CozaLozaWoza {
  2. **public** **static** **void** main(String[] args) {
  3. **int** lowerbound = 1;
  4. **int** upperbound = 110;
  5. **for** (**int** number = lowerbound; number <= upperbound; number++) {
  6. // Print "Coza" if number is divisible by 3
  7. **if** (......) {
  8. System.out.print("Coza");
  9. }
  10. // Print "Loza" if number is divisible by 5
  11. **if** (......) {
  12. System.out.print(.....);
  13. }
  14. // Print "Woza" if number is divisible by 7
  15. ......
  16. // Print the number if it is not divisible by 3, 5 and 7
  17. **if** (......) {
  18. ......
  19. }
  20. // Print a space
  21. ......
  22. // Print a newline if number is divisible by 11
  23. **if** (......) {
  24. System.out.println();
  25. }
  26. }
  27. }
  28. }

1. Fibonacci

In mathematics, the Fibonacci numbers are the numbers in the following integer sequence:

0,\;1,\;1,\;2,\;3,\;5,\;8,\;13,\;21,\;34,\;55,\;89,\;144,\; \ldots\;

By definition, the first two numbers in the Fibonacci sequence are 0 and 1, and each subsequent number is the sum of the previous two.

In mathematical terms, the sequence Fn of Fibonacci numbers is defined by the recurrence relation

F_n = F_{n-1} + F_{n-2},\!\,

with seed values[1]

F_0 = 0,\; F_1 = 1

Write a program called **Fibonacci** to display the first 20 Fibonacci numbers F(n),

1. Special Characters and Escape Sequences

Write a program called PrintAnimalPattern, which uses println() to produce this pattern

'\_\_'

(©©)

/========\/

/ || %% ||

\* ||----||

¥¥ ¥¥

Hint:

**"\u00a9"** => ©

**"\u00A5"** => ¥

1. Print Pattern using nested-loop

# # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # #

(a) (b) (c) (d)

Hints: On the diagonal, row = col. On the opposite diagonal, row + col = size + 1.

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

# # # # # # # # # #

# # # # # # # # # #

# # # # # # # #

# # # # # # # # # #

# # # # # # # # # #

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

(e) (f) (g) (h) (i)

# # # # # # # # # # # # #

# # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # # # # #

# # # # # # # # # # # # # # # # # # # # # # #

(j) (k) # # # # # # # # #

# # # # # # #

# # # # #

# # #

#

(l)

1. Print Number using nested-loop

Write a program to print these numbers:

1 1 2 3 4 5 6 7 8 1 8 7 6 5 4 3 2 1

1 2 1 2 3 4 5 6 7 2 1 7 6 5 4 3 2 1

1 2 3 1 2 3 4 5 6 3 2 1 6 5 4 3 2 1

1 2 3 4 1 2 3 4 5 4 3 2 1 5 4 3 2 1

1 2 3 4 5 1 2 3 4 5 4 3 2 1 4 3 2 1

1 2 3 4 5 6 1 2 3 6 5 4 3 2 1 3 2 1

1 2 3 4 5 6 7 1 2 7 6 5 4 3 2 1 2 1

1 2 3 4 5 6 7 8 1 8 7 6 5 4 3 2 1 1

(m) (n) (o) (p)

1 1 2 3 4 5 6 7 8 7 6 5 4 3 2 1

1 2 1 1 2 3 4 5 6 7 6 5 4 3 2 1

1 2 3 2 1 1 2 3 4 5 6 5 4 3 2 1

1 2 3 4 3 2 1 1 2 3 4 5 4 3 2 1

1 2 3 4 5 4 3 2 1 1 2 3 4 3 2 1

1 2 3 4 5 6 5 4 3 2 1 1 2 3 2 1

1 2 3 4 5 6 7 6 5 4 3 2 1 1 2 1

1 2 3 4 5 6 7 8 7 6 5 4 3 2 1 1

(q) (r)

1 1 1 2 3 4 5 6 7 8 7 6 5 4 3 2 1

1 2 2 1 1 2 3 4 5 6 7 7 6 5 4 3 2 1

1 2 3 3 2 1 1 2 3 4 5 6 6 5 4 3 2 1

1 2 3 4 4 3 2 1 1 2 3 4 5 5 4 3 2 1

1 2 3 4 5 5 4 3 2 1 1 2 3 4 4 3 2 1

1 2 3 4 5 6 6 5 4 3 2 1 1 2 3 3 2 1

1 2 3 4 5 6 7 7 6 5 4 3 2 1 1 2 2 1

1 2 3 4 5 6 7 8 7 6 5 4 3 2 1 1 1

(s) (t)

1

2 3 2

3 4 5 4 3

4 5 6 7 6 5 4

5 6 7 8 9 8 7 6 5

6 7 8 9 0 1 0 9 8 7 6

7 8 9 0 1 2 3 2 1 0 9 8 7

8 9 0 1 2 3 4 5 4 3 2 1 0 9 8

(u)

1. Pascal Triangle

Write a program to print pascal triangle:

1

1 2 1

1 2 4 2 1

1 2 4 8 4 2 1

1 2 4 8 16 8 4 2 1

1 2 4 8 16 32 16 8 4 2 1

1 2 4 8 16 32 64 32 16 8 4 2 1

1 2 4 8 16 32 64 128 64 32 16 8 4 2 1

(a) PowerOf2Triangle

1 1

1 1 1 1

1 2 1 1 2 1

1 3 3 1 1 3 3 1

1 4 6 4 1 1 4 6 4 1

1 5 10 10 5 1 1 5 10 10 5 1

1 6 15 20 15 6 1 1 6 15 20 15 6 1

(b) PascalTriangle1 (c) PascalTriangle2

1. Random array

Write a program that randomly fills a 3 by 4 by 6 array, then prints the largest and smallest values in the array.

Hint: using Randome Object

1. GCD Program

Exercise (GCD): One of the earlier known algorithms is the Euclid algorithm to find the GCD of two integers (developed by the Greek Mathematician Euclid around 300BC). By definition, GCD(a, b) is the greatest factor that divides both a and b. Assume that a and b are positive integers, and a≥b, the Euclid algorithm is based on these two properties:

GCD(a, 0) = a

GCD(a, b) = GCD(b, a mod b), where (a mod b) denotes the remainder of a divides by b.

For example:

GCD(15, 5) = GCD(5, 0) = 5

GCD(99,88) = GCD(88,11) = GCD(11,0) = 11

GCD(3456,1233) = GCD(1233,990) = GCD(990,243) = GCD(243,18) = GCD(18,9) = GCD(9,0) = 9

The pseudocode for the Euclid algorithm is as follows:

* 1. GCD(a, b) // assume that a ≥ b
  2. while (b != 0) {
  3. // Change the value of a and b: a ← b, b ← a mod b, and repeat until b is 0
  4. temp ← b
  5. b ← a mod b
  6. a ← temp
  7. }
  8. // after the loop completes, i.e., b is 0, we have GCD(a, 0)
  9. GCD is a