Exercise 1: Create a class with a method which can calculate the sum of first n natural numbers which are divisible by 3 or 5.

Method Name	calculateSum
Method Description	Calculate Sum
Argument	int n
Return Type	int-sum
Logic	Calculate the sum of first n natural numbers which are divisible by 3 or 5.

Exercise 2: Create a class with a method to find the difference between the sum of the squares and the square of the sum of the first n natural numbers.

Method Name	calculateDifference
Method Description	Calculate the difference
Argument	int n
Return Type	int - Sum
Logic	Find the difference between the sum of the squares of the first n natural numbers and the square of their sum. For Example if n is 10,you have to find (1^2+2^2+3^2+....9^2+10^2)- (1+2+3+4+5...+9+10)^2

Exercise 3: Create a class containing a method to create the mirror image of a String. The method should return the two Strings separated with a pipe(|) symbol .

Method Name	getImage
Method Description	Generate the mirror image of a String and add it to the existing string.
Argument	String
Return Type	String
Logic	Accepts One String Find the mirror image of the String Add the two Strings together separated by a pipe(|) symbol. For Example Input : EARTH Output : EARTH|HTRAE Hint: Use StringBuffer API (Ex: For this problem reverse method in Stringbuffer can be used) Note: Learn the other APIs in StringBuffer

Exercise 4: Create a method to check if a number is an increasing number

Method Name	checkNumber
Method Description	Check if a number is an increasing number
Argument	int number
Return Type	boolean
Logic	A number is said to be an increasing number if no digit is exceeded by the digit to its left. For Example : 134468 is an increasing number

Example 5: Create a method to check if a number is a power of two or not

Method Name	checkNumber
Method Description	Checks if the entered number is a power of two or not
Argument	int n
Return Type	boolean
Logic	Check if the input is a power of two. Ex: 8 is a power of 2

Example 6: A school offers medals to the students of tenth based on the following criteria

If(Marks>=90) : Gold

If(Marks between 80 and 90) : Silver

If(Marks between 70 and 80) : Bronze

Note: Marks between 80 and 90 means → marks>=80 and marks<90

Write a function which accepts the marks of students as a Hashmap and return the details of the students eligible for the medals along with type of medal.

The input hashmap contains the student registration number as key and mark as value.

The output hashmap should contain the student registration number as key and the medal type as value.

Method Name	getStudents
Method Description	Generate the list of students eligible for scholarship
Argument	Hashmap
Return Type	Hashmap
Logic	The method should return the details of the students eligible for the medals along with the medal type.

Example 7: Create a method which accepts a String and replaces all the consonants in the String with the next alphabet.

Note: Consonant refers to all alphabets excluding vowels

Method Name	alterString
Method Description	Replace consonants
Argument	String
Return Type	String
Logic	Return the String replacing all the consonants with the next character. For Example :JAVA should be changed as KAWA

Example 8: Create a method which accepts an array of integer elements and return the second smallest element in the array

Method Name	getSecondSmallest
Method Description	Get the second smallest element in the array
Argument	int[]
Return Type	int

Logic	Sort the array and return the second smallest element in the array
	Hint:
	    1. Convert to ArrayList
	    2. Use sort method in Collections class

Example 9: Create a method which can perform the following operations on two String objects S1 and S2. The output of each operation should be added to an arraylist and the arraylist should be returned.(Assume S2 is of smaller size)

Examples for below statements are shown in the Logic part

1. Character in each alternate index of S1 should be replaced with S2
2. If S2 appears more than once in S1, replace the last occurrence of S2 in S1 with the reverse of S2, else return S1+S2
3. If S2 appears more than once in S1, delete the first occurrence of S2 in S1, else return S1
4. Divide S2 into two halves and add the first half to the beginning of the S1 and second half to the end of S1.
   Note: If there are odd number of letters in S2, then add (n/2)+1 letters to the beginning and the remaining letters to the end. (n is the number of letters in S2)
5. If S1 contains characters that is in S2 change all such characters to *

Method Name	modifyStrings
Method Description	Perform the above mentioned actions on a String
Argument	String,String
Return Type	Arraylist

Logic	Do the above mentioned actions on the entered String.
	For Example
	S1="JAVAJAVA"
	S2="VA'
	    1. **VA**A**VA**A**VA**A**VA**A (J replaced with VA, V replaced with VA etc.)
	    2. JAVAJA**AV**
	    3. JAJAVA
	    4. **V**JAVAJAVA**A**

	5. J***J*** **Output:**{" **VA**A**VA**A**VA**A**VA**A"," JAVAJA**AV**"," JAJAVA"," **V**JAVAJAVA**A**","J***J***"}

Example 10: Create a method that accepts a number and modifies it such that the each of the digit in the newly formed number is equal to the difference between two consecutive digits in the original number. The digit in the units place can be left as it is.

Note: Take the absolute value of the difference. Ex: 6-8 = 2

Method Name	modifyNumber
Method Description	Accepts a number and modify it as per the requirement
Argument	int number1
Return Type	int

Logic	Accept a number and modify it such that the each of the digit in the newly formed number is equal to the difference between two consecutive digits in the original number.
	For example.
	Input: 45862
	Output:13242
	Algorithm:
	    • Convert number into String
	    • Extract each char using charAt method
	    • Convert char to int and find the difference
	    • Create new StringBuffer object and keep adding the difference
	    • Finally convert StringBuffer to int

Example 11: Create a method which accepts the date of birth of person and date format and print the day (SUNDAY, MONDAY…) on which he was born.

Note: The output should be in upper case

Method Name	getDayofWeek
Method Description	Finds the day of the week in which a person is born
Argument	String date, String dateFormat
Return Type	String – Day of week

Logic	Use Calendar API and switch case to get the day of the week
	Ex: Input1 = 25/06/2012
	     Input2 = dd/MM/yyyy
	     Output= MONDAY

Example 12: You are asked to create an application for registering the details of jobseeker. The requirement is:

Username should always end with **_job** and there should be atleast minimum of 8 characters to the left of **_job**. Write a function to validate the same. Return true in case the validation is passed. In case of validation failure return false.

Method Name	validateUserName
Method Description	Checks if the username is valid
Argument	String userName
Return Type	boolean
Logic	Checks if the username ends with _job and contains at least 8 characters to the left of _job. If valid return true. Else return false.

Example 13: Create a method that can accept an array of String objects and sort in alphabetical order. The elements in the left half should be completely in uppercase and the elements in the right half should be completely in lower case. Return the resulting array.

Note: If there are odd number of String objects, then (n/2)+1 elements should be in UPPPERCASE

Method Name	getArrayList
Method Description	Converts the String array to ArrayList and sorts it
Argument	String []elements
Return Type	String [] modifiedArray
Logic	Load the elements in to an ArrayList ,sort it, convert the left half element to uppercase and right half elements to lower case . Hint : 1. Use Collection 2. Use String API

Example 14: Create a method which can remove a List from another List

Method Name	removeElements
Method Description	Removes the elements in one list that is present in the second list also.
Argument	List<String> list1, List<String> list2;
Return Type	List- ArrayList contains the resulting List after the removal process.
Logic	Accept two List objects list1 and list2 and remove the elements from list1 that are present in list2. This should be done in single step process without using loop. Hint: Use the List API which removes all the items in List1 which are contained in List2

Example 15: Create a method which can remove all the elements from a list other than the list of elements specified.

Method Name	removeElements
Method Description	Remove all the elements from a list other than the list of elements specified.
Argument	List<String> list1, List<String> list2;
Return Type	List- ArrayList contains the resulting List after the removal process.
Logic	Accept two List objects list1 and list2 and remove all the elements from list 1 other than the elements contained in list2.This should be done in single step process without using loop. **Hint**: Use the List API method which can retain the elements available in the second list only

Example 16: Create a method which accepts an array of numbers and returns the numbers and their squares in an HashMap

Method Name	getSquares
Method Description	Accepts a list of numbers and return their squares
Argument	int[]
Return Type	Map
Logic	Iterate through the list, find the square of each number and add the elements to a map object with the number as the key and the square as the value.

Example 17: Create a method which accepts the id and the age of people as a Map and decide if they are eligible for vote. A person is eligible for vote if his age is greater than 18. Add the IDs of all the eligible persons to list and return the list. (Assume date is in DD/MM/yyyy format)

Method Name	votersList
Method Description	Generate the list of voters based on the ages of the people
Argument	Map
Return Type	List
Logic	Accept a map with ID as key and Date of Birth as value and check if the person is eligible to vote. A person is eligible for vote for if his age is greater than 18. If the person is eligible add his ID to the list. Hint: Use Calendar API and SimpleDateFormat

Example 18: Create a method which accepts an integer array, reverse the numbers in the array and returns the resulting array in sorted order

Method Name	getSorted
Method Description	Return the resulting array after reversing the numbers and sorting it
Argument	int []
Return Type	int

Logic	Accept and integer array, reverse the numbers in the array, sort it and return the resulting array. Hint
	    1. Convert the numbers to String to reverse it
	    2. Use Collection APIs to sort it
	Ex: {12,23,96,45}
	Step 1: Reverse numbers
	{21,32,69,54}
	Step2: Sort it
	{21,32,54,69}
	Hint: Use String to reverse number
	To sort it, Convert array to ArrayList and use Collections.sort

Example 19: Create a method which accepts an integer array and removes all the duplicates in the array. Return the resulting array in descending order

Method Name	modifyArray
Method Description	Remove duplicates
Argument	int []
Return Type	int []

Logic	Remove the duplicate elements in the array and sort it in descending order
	Hint:
	    1. Use Collection API (TreeSet) to remove duplicates and sort the result in ascending order
	    2. Create a new array, iterate through elements in TreeSet and add it in the reverse order

Example 20: Create a method that accepts a character array and count the number of times each character is present in the array. Add how many times each character is present to a hash map with the character as key and the repetitions count as value

Method Name	countCharacter
Method Description	Count the number of occurrence of each character in a Character array
Argument	char[]
Return Type	map
Logic	Count the number of times each character appears in the array. Add the details into a hash map with character as key and count as value. Example: {'A','P','P','L','E'} Output: Will be hashmap with the following contents{'A':1,'P':2,'L':1,'E':1}

Example 21: A String contains a list of states and capitals. Write a method which can parse the string and return the states and capitals as map with state as key and capital as value.

The String is in the below format.

The state and capital is separated by a delimiter (del1). There will be multiple state-capital pairs and each state – capital pair is separated by another delimiter (del2).

Ex: Input will be **tamilnadu||chennai-karanataka||bengaluru.**

Here, || will be provided as del1 and - will be provided as del2.

Method Name	getStates
Method Description	Accepts the states and capitals as a String and return a map
Argument	String data, char del1,char del l2
Return Type	Map
Logic	Parse the string based on the delimiters and load it to a map with the state name as key and capital as value. Hint: Use Stringtokenizer or split method in String class. Try both the above ways to get familiarized with both APIs

Example 22: In a certain television game show, a couple is considered as a perfect couple if both the husband's and wife's name contains the same set of characters. Each couple is provided with an ID. Write a method which can accept a Hashmap with ID as key and the husband's and wife's name separated with "-" as value. The method should generate the list of perfect couples based on the above mentioned criteria and return their IDs as List object.

Method Name	checkPerfectCouple
Method Description	Select the set of perfect couples
Argument	Map
Return Type	List
Logic	Accept the Map Iterate through it Separate the husband's and wife's names If they contain the same characters, add the ID to the List object. Ex: Assuming VIMAL-MALIV is the value, this is a perfect couple since both these names contains same characters (in different order).

Example 23: Create a method which can perform a particular String operation based on the user's choice. The method should accept the String object and the user's choice and return the output of the operation.

Options are

A: Add the String to itself

B: Replace alternate positions with *

C: Remove duplicate characters in the String

D: Change alternate characters to upper case

Method Name	changeString
Method Description	Modify the string based on user choice
Argument	String string, char ch
Return Type	String
Logic	Perform the required operation based on the user choice and return the resulting string

Example 24: Create a method that accepts a String and checks if it is a positive string. A string is considered a positive string, if on moving from left to right **each** character in the String comes after the previous characters in the Alphabetical order.

For Example

ANT is a positive String (Since T comes after N and N comes after A)

APPLE is not positive since L comes before P in the alphabetical order.

The method should return true if the entered string is positive

Method Name	checkPositive
Method Description	Checks if a String is positive
Argument	String
Return Type	boolean
Logic	Check if a string is positive based on the above criteria and return true if positive. Hint: **Step 1:** Convert to Char array. **Step 2:** Iterate through array, subtract 1st two characters (A-N). This will give the ASCII difference **Step 3:** If result is negative, then return false and break. Else continue to next loop

Example 25: Create a method which accepts two Arraylist containing characters. Merge both arrays lists, sort the elements in the resulting list and return the resulting array.

Method Name	mergeData
Method Description	Merge two arraylist , sort it and return the result as an integer array.
Argument	List, List
Return Type	char[]
Logic	Merge both arrays lists, sort the elements in the resulting list and return it as a char array.

Example 26: Create a method that searches for a particular String in a List. If found, the element should be replaced with a string having only half of the characters in the actual string

Method Name	modifyElement
Method Description	Search for an element in the arraylist and modifies it.
Argument	List<String> arrayList , String element
Return Type	List

Logic	Accept an arraylist and search for an element in the list and replace with a string having only first half of the characters in the actual string. For Example if a search was done for APPLE and if APPLE is found in the list, replace it with APP. Return the modified list Hint: Iterate through list and find the index where the String is present. Take the first half of the String and set it at that index in the arraylist. (Use set method)

Example 27: Create a method to find the sum of the first n even numbers that are divisible by 3 and 8

Method Name	findSum
Method Description	Find the sum of first n even numbers that are divisible by 3 and 8
Argument	Int
Return Type	Int
Logic	Sum of the multiples of first n even numbers that are divisible by 3 and 8

Example 28: Create a method to find the sum of the cubes of the digits of an n digit number

Method Name	findSum
Method Description	Find the sum cubes of the digits of an n digit number
Argument	Int
Return Type	Int
Logic	Return the sum of cubes of the digits of an n digit number Example Input : 123 Output : 1^3+2^3+3^3= 1+8+27=36 Hint: Use %(mod) operator to separate each digit

Example 29: Create a method which accepts a hash map and return the values of the map in sorted order as a List.

Method Name	getValues
Method Description	Get the values of a map in sorted order
Argument	HashMap
Return Type	List
Logic	Return the values of a hash map in sorted order

Example 30: A company requires each employee to maintain a secret code. The secret code needs to pass certain validation for getting accepted.

The validation rules are as given

1. The secret code should be six characters long
2. The first three characters should be cod (Use .startsWith method)
3. There should be at least one digit in the code (Use .isDigit)
4. The first character should always be an upper case letter(Use isUpperCase)
5. The code should contain only alphabets and digits.
6. The number of upper case letters should be greater than lower case letters.

Return true if the above validation is passed.

Method Name	validateCode
Method Description	Validate the entered code as per the given validation rules
Argument	String code
Return Type	boolean
Logic	Validate the entered code Hint: Use the String API methods to extract each character

Example 31: Write a method to find the sum of the factorials of the first n numbers in the Fibonacci series.

Fibonacci series is given by 0 1 1 2 3 5 8……

Factorial for a number m is given by factorial= m*m-1*m-2….1

Method Name	sumOfFactorial
Method Description	Calculate sum
Argument	int n
Return Type	int
Logic	1. Generate the first n elements in the Fibonacci series 2. Find the factorial of each element 3. Find the sum of the factorial

Example 32: A company transmits its String data over the network as encrypted data. The encryption logic is as shown given below.

For a String ad the logic is as given

a→ a+9=j

d→d+9=m

So the encrypted word would be jm.

If on addition of 9 results in a char greater than z (ASCII value 122) do the encryption in a cyclic manner starting from a. So if any character is 'z' it should be (z+9) which is equal to 127>122. In this case the character would be 9 character starting from 'a' which 'i' so for adz the encrypted value should be adi

Method Name	encryptString
Method Description	Encrypt the entered string
Argument	String
Return Type	String
Logic	1. Perform the arithmetic operation of char data. 2. For example Assume char a='b'; a++; Now the value of a will be c. This is

	because in java the arithmetic operation on character works on its ASCII value. The ASCII value of 'a' is 97 and that of 'z' is 122.

Example 33: A sales company keeps track of the product purchased and sold. The company needs to make sure that the sale date is always after the purchase date. Write a method to verify this

Method Name	compareDates
Method Description	Comparing the purchase date and selling date
Argument	String purchaseDate, String sellingDate
Return Type	boolean
Logic	1. Convert the string to Date objects 2. Return true if the selling date comes after the purchase date

Example 34: A company used to keep the record of the employees in two different branches separately. There are some employees who work in both the location. The company needs to keep track of the employee working in both the branches. Write a method to accept the two lists containing the names of the employees working in the two branches. The method should find out the names of employees present in both the list and return the names as a sorted array

Method Name	getEmployees
Method Description	Get the names of employees working two different branches
Argument	List branch1, List branch2
Return Type	String []
Logic	Find the common names of the employees in both the lists

Example 35: In a school there are some teachers who handle two subjects (Maths and English). When the feedback was taken their feedback was present in both Maths Feedback as well as English Feedback. Write a method to create a consolidated feedback for the teachers for English and Maths. For those taking both the subjects the highest feedback is taken. Write a method to accept two maps and return a Map object containing the feedbacks of all teachers in maths and English.

Note: HashMap contains

- Key - Teacher's name of type String
- Value – Feedback of type Integer

Method Name	getConsolidateFeedback
Method Description	Get the consolidated feedback
Argument	Map englishFeedback, Map mathsFeedBack
Return Type	Map consolidatedFeedBack
Logic	Hint : 1. Extract the keys of both the maps 2. For the teachers handling both the subjects find their maximum feedback and add to the output map. 3. Add the feedbacks of other teachers into a new Map

Example 36: Write a method which can find the sum of the first n prime numbers. Prime numbers are numbers which have only 1 and the number itself as its factors. 2 is the only even prime number. 1 is neither prime nor composite.

Ex: the 1st 5 prime numbers are 2,3,5,7,11 and sum is 28

Method Name	getPrimeSum
Method Description	Get the sum of the first n prime numbers
Argument	int n
Return Type	Int
Logic	**Hint** : 1. Use for loop to iterate over numbers from 2 to n, say loop variable i. 2. Use an inner loop with loop variable j which loops from to 2 to i/2. If for any j the remainder on dividing i by j is zero, the number is non-prime. If it is prime add the number to the sum.

Example 37: Write a method which accepts a String and moves all the lower case 'a' to the beginning of the String.

Method Name	rearrangeCharacters
Method Description	Move the all the lower case 'a' to the beginning of a String
Argument	String
Return Type	String

Logic	Hint :
	1. Convert the string to a character array
	2. Create a Stringbuffer object
	3. Create a variable(**count**) to store the number of 'a' present
	4. Iterate over the character array and if the character is 'a' increment **count** for 'a' else add the character to the StringBuffer object.
	5. Finally insert the **count** number of 'a' to the beginning of the StringBuffer object

Example 38: Write a method which can find the factors of a number. Factor is an integer which evenly divides a number without leaving a remainder. Return the factors as an arraylist object.

For Example 1, 2 and 4 are the factor of 4

Method Name	getFators
Method Description	Get the factors of a number n
Argument	int n
Return Type	List
Logic	Hint :
	1. Create a loop starting from 1 to n with loop variable say i.
	2. Check if for any i , dividing n by i gives zero as remainder. Then i is a factor of n .
	3. Add i to the list object

Example 39: Write a method which can accept an integer and return the binary, hexadecimal and octal equivalents of the number in a String array

Method Name	getFormats
Method Description	Gets the binary,hexadecimal and octal formats of a number
Argument	int
Return Type	String
Logic	Hint :
	1. Use Integer wrapper class methods

Example 40: Write a method which accepts a double number and finds the sum of the digits to the left and right of the decimal point. It should return the sum as String in the following format

Left side sum:Right side sum

For example

Input :120.520

Output: 3:7

Method Name	getSum
Method Description	Get the sum of digits on either sides of the decimal points in a double number
Argument	double
Return Type	String
Logic	Hint : 1. Convert the double number to aString 2. Separate the String to two parts based on the decimal point. 3. Find the sum of digits on each part

Example 41: Write a method to validate the age of a person. The person age is considered valid if it is above 21 years. Accept the date of birth of the person as String in date-month-year(Ex: 23-05-2012) format and return true if the age is greater than 21.

Method Name	validateAge
Method Description	Validates if the person's age is above 21
Argument	String
Return Type	Boolean
Logic	Hint :Use Calendar, Date and DateFormat APIs

Example 42: Write a method which can return the current date in any of the following date formats based on the user choice

Choice 1 : Month-date-year(Ex: 05-26-2012)

Choice 2: Date-month-year(Ex: 26-MAY-12)

Choice 3: Date/month/year(Ex: 26/05/2012)

Choice 4: Month/date/year(Ex: 05/26/2012)

Choice 5: Return the current year

Choice 6: Return the current month as (Ex: APR)

Choice7: Return the date 10 days after the sysdate

Choice 8: Return 10 days prior to sysdate

Method Name	getDate
Method Description	Format the entered date in the specified formats
Argument	int
Return Type	String
Logic	Use switch case(For choice) and DateFormat API

Example 43:

Consider two Hashmaps .First one containing the product name and product category code as key and value respectively. Second HashMap contains the product name and the units sold. Write a java function which accepts the two hash maps and return the names of products in each category which is having the highest number of units sold.

For example

Input1 :{"lux":"soap","colgate":"paste", "pears":"soap","sony":"electronics","samsung":"electronics"}

Input 2:{"lux":1000,"colgate":500,"pears":2000,"sony":100," samsung",600}

Output: {"pears","colgate","samsung"}

Method Name	getMaxSold
Method Description	Find the product maximum sold in each category
Argument	HashMap<String,String> productDetails, HashMap<String,Integer> salesDetails
Return Type	List
Logic	Algorithm: 1. Get product names from the productDetails map using the getValues method(Say categoryName which will be of data type Collection. 2. Remove the duplicate category names by converting categoryName to a Set object which can be done using the constructor of HashSet which accepts Collection object as argument

	3. Obtain an iterator object(**iterator1**) to iterate over the category names and iterate over the set 4. Get first element in the set (**category**). 5. Declare two variables to maxSaleCount and maxSaleProduct with initial values 0 and null. 6. Inside the iteration create another iterator (**iterator2**) to iterate over the keys of the salesDetails map. 7. Iterate iterator2 and get the first element(**product**) 8. Check if the value for the key product in the productDetails map is equal to category. If found equal compare the sales of the product with the maxCount value and if the sales>maxCount set maxCount as sales and maxSaleProduct as **product.** 9. Completing the **iterator2** once will give the product which was sold maximum in a particular category 10. Completion of **iterator1** gives the names of the products which was sold maximum in each categories.

Example 44: Write a method which accepts a number and return it in words.

For Example 123 → One Two Three

Method Name	getNumber
Method Description	Get the number in words
Argument	int
Return Type	String
Logic	Use mod(%) operator, StringBuffer and switch case

Example 45: Consider two Hashmaps .First one containing the product name and product category code as key and value respectively. Second HashMap contains the product name and price. Write a java function which accepts the two hash maps , price hike rate and the product category and updates the prices of the product in the entered category by the hike rate

For example

Input1 :{"lux":"soap","colgate":"paste", "pears":"soap","sony":"electronics","samsung":"electronics"}

Input 2:{"lux":1000,"colgate":500,"pears":2000,"sony":100," samsung",600}

Input 3:10

Input4: "electronics"

Output1: :{"lux":1000,"colgate":500,"pears":2000,"sony":110," samsung",660}

Method Name	updatePrices
Method Description	Get the number in words
Argument	HashMap<String,String> productDetails, HashMap<String,Integer> salesDetails, int rate, String category
Return Type	String
Logic	Hint 1: 1. Iterate over the productDetails Map and get the names of the products in the entered category. The product names can be added to a Set 2. Iterate over the set whose elements will be present as key in the salesDetails tables. For each productEntry in the set update the values of the salesDetails map with rate%.

Example 46: Write a method which can check whether an entered number is palindrome or not.

Method Name	checkPalindrome
Method Description	Check palindrome
Argument	int
Return Type	boolean
Logic	Hint 1: 1. Convert the number to String 2. Check if the String and reverse of the String are equal