Object Oriented Programming Lab Manual(PCC CS593)

Java

Index:

I. Basic Introduction and Input Format(Scanner class and Command Line Argument),Arrays.

Q1. Write a Java program to print "hello World".

Q2. Write a Java program to check whether a string is palindrome or not using command line argument.

Q3.Write a Java program to find factorial of a number using Scanner class.

Input/Output based questions.

Array:

1)

Write a java program to create an array of size ‘n’ and accepts ‘n’ number of integers from the user and display all the values from an array in reverse order

Sample Input 1

Enter the array size

3

Enter the values

46

83

67

Sample Output 1

67

83

46

2)

Write a java program to find the given number from the array of elements and display its position. If the number is not present in an array then display it as 0.

Assume the position starts from 1.

Sample Input 1

Enter the array size

4

Enter the values

9

32

17

4

Enter the number to find

17

Sample Output 1

3

3)

Write a java program to sort the values from the array of elements in ascending order

Sample Input 1

Enter the array size

4

Enter the values

16

3

77

83

Sample Output 1

3

16

77

83

4)

Write a java program to sort the values from the array of elements in descending order

Sample Input 1

Enter the array size

5

Enter the values

26

10

416

92

7

Sample Output 1

416

92

26

10

7

5)

Anjali gets n numbers in an array. Write a Java program to print the sum of the maximum and the minimum element in the array.If the size of an array is 0 or less print "Invalid Array Size".

Sample Input 1:

Enter the size of an array:

5

Enter the elements:

45

23

48

90

89

Sample Output 1:

113

Sample Input 2:

Enter the size of an array:

0

Sample Output 2:
Invalid Array Size

6)
Interchange the elements of an array

Write a Java program to interchange the elements of an array with the elements of another array without using the third array.If the array size differs display "Unable to swap size differs".If the range is lesser or equal to Zero . Display "Invalid range "

Assume the maximum size of array is 20

Sample Input 1:
Enter the number of elements in the first array :
3
Enter the elements in the first array
1
2
3
Enter the number of elements in the second array :
3
Enter the elements in the second array

4
5
6

Sample Output 1:

The first array after swapping is :
4 5 6
The second array after swapping is :
1 2 3

Sample Input 2:

Enter the number of elements in the first array :

-3

Sample Output 2:

Invalid range

Sample Input 3:

Enter the number of elements in the first array :

3

Enter the elements in the first array

1

2

3

Enter the number of elements in the second array :

2

Sample Output 3:

Unable to swap size differs

7)

Sum of factorial of positive and single digit numbers in an array

Write a java program to find the sum of factorial of the numbers in an array. Consider the number for finding the factorial only if it is positive and single digit. If not print "No positive and single digit numbers found in an array".

Example if the array is {2,-7,14,-24,41,5} the output should be 122

Sample Input 1:

Enter the size of an array:

5

Enter the elements:

2

-56

-13

6

56

Sample Output 1:

722

8)

Sort the first and second half of an array

Anjali likes to play mathematical tricky games .She gets n numbers for an array. Help Anjali to write a Java program to sort the first half of the array in ascending order and the second half of the array in descending order. If the size of the array is 0 or lesser then display the message as "Array size should be greater than 0".

Sample Input 1:

Enter the size of an array:

5

Enter the elements:

89
23
56
12
99

Sample Output 1:

23

56

89

99

12

Sample Input 2:

Enter the size of an array:

0

Sample Output 2:
Array size should be greater than 0

9)
Array Compatibility

Two arrays are said to be compatible if they are of the same size and if the ith element in the first array is greater than or equal to the ith element in the second array for all i elements.If the array size is zero or lesser then display the message "Invalid array size".Write a Java program to find whether 2 arrays are compatible or not.If the arrays are compatible display the message as "Arrays are Compatible" ,if not then display the message as "Arrays are Not Compatible".

Sample Input 1:

Enter the size for First array:

5

Enter the elements for First array:

5

14

17

19

15

Enter the size for Second array:

5

Enter the elements for Second array:

2

5

9

15

7

Sample Output 1:

Arrays are Compatible

Sample Input 2:

Enter the size for First array:

3

Enter the elements for First array:

1

4

7

Enter the size for Second array:

5

Enter the elements for Second array:

2

5

9

5

7

Sample Output 2:

Arrays are Not Compatible

Sample Input 3:

Enter the size for First array:

```
-2
```

Sample Output 3:

```
Invalid array size
```

10)

Numerology

Write a program to find the numerological value for a given name.
Note: Store the numerological number and the corresponding character in a 2-D array(2*26). Always the given name should be in capital case ,else the name is not valid. Check for the valid name,if the name is invalid print the message "Invalid name".There should not be any space in the name provided.
For example:
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
1 2 3 4 5 8 3 5 1 1 2 3 4 5 7 8 1 2 3 4 6 6 6 5 1 7

Sample Input 1:

```
Enter your name:
SUDHA
```

Sample Output 1:

```
Your numerology no is:19
```

Sample Input 2:

```
Enter your name:
kiran
```

Sample Output 2:

```
Invalid name
```

Sample Input 3:

```
Enter your name:
ANI34
```

Sample Output 3:

```
Invalid name
```

II. Introduction to class and object concept, method overloading, constructor overloading.

Q4. Write a Java program to implement push and pop operation of the Stack.

Q5. Write a Java program to implement method Overloading and constructor overloading:

i)Method Overloading: class name: Box
class Boxs' members: Box constructor- Box()
: Methods- volume() :no parameter
volume(3 different parameters)
volume(1 parameter)

ii)Constructor Overloading: class name: Box
class Boxs' members: Method- volume()
: Constructors- Box() :no parameter
Box(3 different parameters)
Box(1 parameter)

Input/Output based questions.

Calculate Years of Experience

Get the date of joining(as String) as input from the user. Calculate the number of years of experience and display the same.

Assume

1) the given date is a valid date and the format is dd/MM/yyyy.

2) The current date is 15/12/2020

Sample Input 1

21/8/2006

Sample Output 1

14 years

Hint : Create two Date objects- one for the input date and other for the current date. (Say d1 and d2). Find the difference between these two dates in milliseconds as

d2.getTime() - d1.getTime().

Convert this milliseconds to number of years

2)

Create a class Person with the following private member variables

- String name
- char gender
- int age

Include appropriate getters and setters method
Create a class BusTicket with the following private member variables

- int ticketNo
- float ticketPrice
- float totalAmount
- Person person

Include appropriate getters and setters method

Write the following method in the BusTicket class
void calculateTotal()—this method should calculate the total and set it based on the discount given below:

1. For Children whose age is less than 16, give 50% discount
2. For Senior citizen whose age is greater than or equal to 60 give 25% discount
3. For Ladies give 10% discount
4. No discount for remaining category.

Based on above condition calculate total price.
Create TestMain class which has main method and do the following input and output process .

To get the input write a method

public static BusTicket getTicketDetails() - Get the inputs relevant to BusTicket in this method and return the BusTicket object. Call this method in the main method, using the returned object invoke the method calculateTotal and print the output accordingly

Note: Do not create a static reference for the Scanner class, create a scanner object inside the method.

Sample input 1:

```
Enter the passenger name:
vivek
Enter the gender(M or F / m or f):
M
Enter the age:
16
Enter the ticket no:
123
Enter the ticket price:
100.0
```

Sample Output 1

```
Ticket no:123
Passenger Name:vivek
Price of a ticket : 100.0
Total Amount : 100.0
```

Sample input 2:

```
Enter the passenger name:
Priya
Enter the gender(M or F / m or f):
F
Enter the age:
61
Enter the ticket no:
140
Enter the ticket price:
500.0
```

Sample Output 1

```
Ticket no:140
Passenger Name:Priya
Price of a ticket : 500.0
Total Amount : 375.0
```

3)

Ticket Price Calculation - Static

Ticket Calculation

Create a class Ticket with the following private variables
int ticketid;
int price;
static int availableTickets;

Include getters and setters methods in the Ticket class.

AvailableTickets should hold only positive value. Zero and negative values are not allowed.(This logic should be checked inside the corresponding setter method)

Write the following method in the Ticket class:

public int calculateTicketCost(int nooftickets) —this method should check the ticket availability, If the tickets are available, reduce the nooftickets from availableTickets and calculate the total amount as nooftickets*price and return the total amount. If the tickets are not available, this method should return -1.

Write a main method in the Main class to test the application.

Note: Avoid using System.exit

Sample input and output

```
Enter no of bookings:
2
Enter the available tickets:
25
Enter the ticketid:
123
Enter the price:
100
Enter the no of tickets:
5
Available tickets: 25

Total amount:500

Available ticket after booking:20

Enter the ticketid:
124
Enter the price:
100
Enter the no of tickets:
2
Available tickets: 20

Total amount:200

Available ticket after booking:18
```

Sample input and output 2:

```
Enter no of bookings:
1
Enter the available tickets:
25
Enter the ticketid:
123
Enter the price:
100
Enter the no of tickets:
26
Tickets not sufficient / available
```

4)

Student Details - Constructor

Create a class Student with the private attributes

int studentId

String studentName, studentAddress, collegeName.

Include appropriate getter methods.

Assume most of the students are from “NIT” college. So user has to give input whether the student is from NIT or not.

1. If student belongs to NIT, give input as 'yes/YES' and skip input for the attribute collegeName and create student object with 3-argument constructor to initilze the values for studentId, studentName and studentAddress and collegeName as “NIT”.
2. If student belongs to other college, give input as 'no/NO' and get college name from the user and create student object with 4-argument constructor to initialize all the values.
3. Instead of Yes / No, if user enters different input then display 'Wrong Input' and get the input again.

Based on the above assumptions write the necessary constructors in the Student class.

Write a class StudentMain with the main method and test the application.

Sample Input 1:

Enter Student's Id:

12

Enter Student's Name:

John

Enter Student's address:

Chennai

Whether the student is from NIT(Yes/No):

NO

Enter the college name:

SVS

Sample Output 1:

Student id:12

Student name:John

Address:Chennai

College name:SVS

Sample Input 2:

Enter Student's Id:

43

Enter Student's Name:

Tom

Enter Student's address:

Coimbatore

Whether the student is from NIT(Yes/No):

y

Wrong Input

Whether the student is from NIT(Yes/No):

yes

Sample Output 2:

Student id:43

Student name:Tom

Address:Coimbatore

College name:NIT

5)

Date Validation - Use SimpleDateFormat

Date Validation

Get a date as String from user. Check if that date is a valid date using SimpleDateFormat.

Date entered by the user is in the format dd/MM/yyyy

If the given date is valid then display "<Date> is a valid date". Else display "<Date> is not a valid date".

Sample Input 1

31/08/2020

Sample Output 1

```
31/08/2020 is a valid date
```

Sample Input 2

```
31/11/2020
```

Sample Output 2

```
31/11/2020 is not a valid date
```

6)

Volume calculator- Over Loading

Sarah got confused to calculate volume of cylinder and cuboid. Write a Java application to help Sarah to do this.

Create a class called VolumeCalculator that has the following methods
double calculateVolume(double radius,double height)
This method calculates the volume of the cylinder using the formula 3.14*radius*radius*height
double calculateVolume(int length,int breadth,int height)
This method calculates the volume of the cuboid using the formula length*breadth*height
Write a TestMain class to test the application.

Sample Input

```
Enter the choice
1.Find Volume For Cylinder
2.Find Volume For Cuboid
1
Enter the radius
3
Enter the height
2
```

Output

```
Volume of the Cylinder is 56.52
```

Sample Input

```
Enter the choice
1.Find Volume For Cylinder
2.Find Volume For Cuboid
2

Enter the length
3
Enter the breadth
2
Enter the height
1
```

Output

```
Volume of the Cuboid is 6.00
```

III. Nested and Inner classes.

Q6. Write down a Java program to show how Nested and Inner classes are used in Java.

IV. Variable Length Arguments (Varargs).

Q7. Write down a program in Java to calculate sum of whole numbers with zero, two and 5 parameters respectively(Use for-each loop).

V. Use of static and final keywords.

Q8. Write down a Java program to view the effect of static and final on instance variables.

VI. Concept of Inheritance and method overriding. Use of super, this and abstract keywords.

Q9. Write a program in Java to implement inheritance:

i) Create a class Box. Extend it using class Rectangle and Square respectively. Override method volume of Box class in both the child classes. use super keyword for constructors.

Q10. In the above program apply' this' and 'super' keywords.

Q11. In the question 9, make the Box class as abstract and then extend the same by using Rectangle class and Square class respectively .

VII. Concept of Interface and polymorphism through dynamic method dispatch.

Q12. Create an interface called Figure and let the abstract method area() be implemented by classes Rectangle and Square. In the main method call the overrided area() method by use of reference variable of interface Figure.

VIII. Concept of Packages.

Q13. Create a class 'Super' under a package called 'pkg'. In this class take for variables of type: default, public, private and protected. Extend this class in the same package by another class called SamePkgChild class and try to access all these variables.

Again import this package 'pkg' and create a child class DiffPkgChild of the 'Super' class; again try to access all the variables of the 'Super' class.

Draw the following table showing the accessibility:

	Private	No Modifier	Protected	Public
Same class	Yes	Yes	Yes	Yes
Same Package Subclass	No	Yes	Yes	Yes
Same Package Non Subclass	No	Yes	Yes	Yes
Different Package Subclass	No	No	Yes	Yes
Different Package Non Subclass	No	No	No	Yes

IX. Concept of Exception Handling(try-catch-finally-throw-throws).

Q14. Write down a Java program with divide by zero error (Don't use any try catch mechanism).

Q15. Write down the above exception generating code in try block and catch the ArithmeticException.

Compare the two output.

Q16. Write a Java program, using command line argument generate errors divide by zero and array index out of bounds exceptions, respectively in a nested try block and catch the same by catch respective blocs .

Q17. In Q15 use finally block after catch.

Q18. Write a Java program to demonstrate use of 'throw' exception handling mechanism.

Q19. Write a Java program to demonstrate use of 'throws' exception handling mechanism (Throw a user defined exception).

Input/Output based questions.

Exception Handling:

1)

Get a number as input from the user using nextInt method of Scanner.

If the input provided by the user is not a number (contains alphabets or special characters), it will throw an Exception, InputMismatchException.

When the input provided by the user is correct, print the number, else handle it using catch block for InputMismatchException and print the message as “Input should be a number”.

Note : InputMismatchException is available in java.util package

Sample Input 1 :

45

Sample Output 1 :

The input is 45

Sample Input 2 :

abc45

Sample Output 1 :

Input should be a number

2)

Calculate Product cost - Try with multiple catch

Calculate Product cost - Try with multiple catch

Get 2 numbers as input from the user using nextInt method of Scanner. First input is the total cost of photo frames (x) and the second input is the number of photo frames purchased (n).

Find the cost of each photo frame (x / n)..

If the input provided by the user is not a number (contains alphabets or special characters), it will result in InputMismatchException. In such case, display the message as "InputMismatchException : Input should be an integer".

Also the second input should not be 0. If so it results in ArithmeticException. In such case, display the message as "ArithmeticException : Cannot divide by zero".

Sample Input 1

Enter the total cost of photo frames

640

Enter the number of photo frames

8

Sample Output 1

Cost of each photo frame is 80

Sample Input 2

Enter the total cost of photo frames

6.5

Sample Output 2

InputMismatchException : Input should be an integer

Sample Input 3

Enter the total cost of photo frames

640

Enter the number of photo frames

ab

Sample Output 3

InputMismatchException : Input should be an integer

Sample Input 4

Enter the total cost of photo frames

640

Enter the number of photo frames

0

Sample Output 4

ArithmeticException : Cannot divide by zero

3)

Display Array element - Usage of finally

Display Array element - Usage of finally

Get 5 names from the user and store it in an array. Get the index position from the user and display the name in that position of the array.

If the index is beyond the range of array, handle it using ArrayIndexOutOfBoundsException and display message as “Index is out of bounds of the array”.

Whether the exception occurs or not, display the message as “Thank you Have a nice day”.

Write a java program for the above requirement.

Sample Input 1 :

John

Arun

Sachin

Dravid

Ashwin

3

Sample Output 1

Dravid

Thank you Have a nice day

Sample Input 2 :

Azhar

Sachin

Pranav

Ashwin

Dravid

5

Sample Output 1

Index is out of bounds of the array

Thank you Have a nice day

4)
Validate Date using ParseException - Use try catch throws

Date Validation using ParseException - Use try catch throws

Get the input for date of birth as String from the user. Check whether the given date is valid using SimpleDateFormat.

Write a java program to meet the above requirement.

Implement throws clause in this requirement.

In the Main class implement the below method

public static Date convertStringToDate(String str) – This method should convert the String str in the format dd/MM/yyyy to java.util.Date using SimpleDateFormat. **This method should throw ParseException**. If valid return the converted Date object.

Note : When using SimpleDateFormat to convert String to Date it will allow the code to compile only if you handle the exception “ParseException” available in java.text package.

Now try to recollect what type of Exception is ParseException – Checked or unchecked ?

In the main method, get the date as String in the format dd/MM/yyyy and invoke the method convertStringToDate. If date is valid display “<String> is a valid date”.

Else handle the Exception thrown and print “<String> is not a valid date”.

Sample Input 1 :

31/10/2020

Sample Output 1

31/10/2020 is a valid date

Sample Input 2 :

31/11/2020

Sample Output 1

31/11/2020 is not a valid date

5)

Display Product Details - User defined Exception

View Product Details – User defined exception

Get the input of product name and price and print the same. The price should be greater than or equal to 100. If valid print the product details. Else throw a user defined exception, InvalidPriceException with the message “InvalidPriceException - Product price invalid”. Print this message using getMessage method.

For this, you are provided with a public class InvalidPriceException. This class should inherit the Exception class and write a single argument constructor (String) which needs to be set to the message of super class.

Use throw keyword to throw user defined exception.

Input consists of a String and an int.

Sample Input 1 :

Enter product name

Perfume

Enter price

360

Sample Output 1

Name : Perfume

Price : 360

Sample Input 2 :

Enter product name

Wallet

Enter price

50

Sample Output 2

InvalidPriceException - Product price invalid

6)
Array Manipulation - Use try with multi catch

Tom wants to store the price details of the products that he purchased from the departmental store. Help him do this by using the concept of Arrays.

To do this create a public class ArrayException with a method getPriceDetails as :

public String getPriceDetails() - This method should do the following

Get the size of an array as input and then get the elements of the array(all elements are int) as input.

Next, user should provide the index of the array. This method should return the element at that index as “The array element is = “+<that value>

This program may generate ArrayIndexOutOfBoundsException / InputMismatchException

In case of ArrayIndexOutOfBoundsException, the function should return “Array index is out of range”.

When providing the input, if the input is not an integer, it will generate InputMismatchException. In this case the function should return “Input was not in the correct format”.

Use exception handling mechanism to handle the exception. Use separate catch block for handling each exception. In the catch block, return the appropriate message.

Write a main method and test the above function.

Sample Input 1:

```
Enter the number of elements in the array
5
Enter the price details
50

80

60

70

40
Enter the index of the array element you want to access
1
```

Sample Output 1:

```
The array element is 80
```

Sample Input 2:

```
Enter the number of elements in the array
2
Enter the price details
50
80
Enter the index of the array element you want to access
9
```

Sample Output 2:

```
Array index is out of range
```

Sample Input 3:

```
Enter the number of elements in the array
2
Enter the price details
30

j
```

Sample Output 3:

```
Input was not in the correct format
```

7)

Divide two numbers - Use finally

Andrew wants to teach division of two numbers to his son. To make his kid understand it better, he wants to write a program that divides two numbers and displays the answer.

Help him do this by writing a java program.

Write a public **class Division**. Write a method divideTwoNumbers as

public String divideTwoNumbers(int number1,int number2) – This method should perform division as number1 / number2.

If number2 is zero, it will throw ArithmeticException.

Whether division is done successfully or not, it should concatenate the String as “Thanks for using the application”.

When division done successfully, it should return a message as

“The answer is <number1/number2>. Thanks for using the application.”

If it results in ArithmeticException, it should return a message as

“Division by zero is not possible. Thanks for using the application.”

Use try, catch and finally to perform the above task.

Write the main method and test the above function.

Sample Input 1:

Enter the numbers

15

3

Sample Output 1:

The answer is 5. Thanks for using the application.

Sample Input 2:

Enter the numbers

15

0

Sample Output 2:

Division by zero is not possible. Thanks for using the application.

Sample Input 3:

Enter the numbers

15

2

Sample Output 1:

The answer is 7. Thanks for using the application.

8)

Register a Candidate - User defined Exception(with throw and throws)

Geneva Technologies is planning to conduct a Walk-in interview. The interview has 4 levels. To attend the interview, the candidates need to register the following information:
Name, Gender and Expected salary.

Help him do this by writing a java program.

Partial code is provided.

You are provided with a public class Candidate with private attributes :

String name

String gender

double expectedSalary

Appropriate getter and setters are provided.

You are provided with a public class Main.

Write a method getCandidateDetails as –

public static Candidate getCandidateDetails() – This method should get the candidate details, create the Candidate object using those details and return that object.

If the candidate’s expected salary is less than 10000

- throw a user defined exception as InvalidSalaryException with the message “Registration Failed. Salary cannot be less than 10000.” and return null.

- this method should throw / propagate InvalidSalaryException.

To do this, write a class InvalidSalaryException that inherits Exception class.

Write a constructor that takes a String as argument and set this string to the message attribute of the super class, Exception.

In the Main class, write the main method and test the method getCandidateDetails.

If it returns a valid Candidate object, then display “Registration Successful”.

Use a catch block to handle the exception that is returned by the method getCandidateDetails. In catch block display the message by using the getMessage() method.

Sample Input 1:

Enter the candidate Details

Name

Margrett

Gender

Female

Expected Salary

50000

Sample Output 1:

Registration Successful

Sample Input 2:

Enter the candidate Details

Name

Robin

Gender

Male

Expected Salary

5000

Sample Output 2:

```
Registration Failed. Salary cannot be less than 10000.
```

X. Concept of Multithreading(Use of Runnable,Thread,sleep(),stop(),yield(),isAlive()-join()).

Q20. Write down a program in java to create two threads -'thread1' and 'thread2' by:

i)extending Thread class

ii)Implementing Runnable Interface.

Q21. In Q 20, use stop() method for 'thread1' and check the output.

Q22. In Q 20, use sleep(3000) method for 'thread2' and check the output.

Q23. Demonstrate the use of yield method by a small Java Program.

Q24. Demonstrate the use of IsAlive() and join() methods by using a small Java program.

Q25. Demonstrate the use of synchronized keyword by a small Java Program.

XI. Applet Programming.

Q26. Write down a program in Java to demonstrate parameter passing in applets.
Q27. An applet program for demonstrating the concept of delegation event model and listener.
Q28. An applet program to understand the following concepts:

I. I/O in applets.
II. use of repaint(), III.
getDocumentBase(), IV.
getCodeBase(),
V. layout manager basic concept, VI.
JButton class,
VII. textFields.

XII. String class.

Q29. Write down a program in Java to demonstrate String object is immutable use concate() method.
Q30. Write down a program in Java to demonstrate use of toUpperCase(), toLowerCase(), startsWith(), endsWith(), charAt(), length(), equals(), replace(), substring().
Q31. Write down a program in Java to demonstrate the mutable string StringBuffer class.
Use the following methods:

VIII. append()
IX. insert()
X. replace()
XI. delete()
XII. reverse()

Input/Output based questions.

Strings in JAVA

1) Write a Java program to get the character at the given index within the String.

Sample Output

```
Original String = Java Exercises!
The character at position 0 is J
The character at position 10 is i
```

2) Write a Java program to compare two strings lexicographically.
(Two strings are lexicographically equal if they are the same length and contain the same characters in the same positions.
Sample Output

```
String 1: This is Exercise 1
String 2: This is Exercise 2
```

```
"This is Exercise 1" is less than "This is Exercise 2"
```

3) Write a Java program to compare two strings lexicographically,
Ignoring case differences

Sample Output

```
String 1: This is exercise 1
String 2: This is Exercise 1
"This is exercise 1" is equal to "This is Exercise 1"
```

4) Write a Java program to concatenate a given string to the end of another string.

Sample Output:

```
String 1: PHP Exercises and
String 2: Python Exercises
The concatenated string: PHP Exercises and Python
Exercises
```

5) Write a Java program to replace each substring of a given string

that matches the given regular expression with the given replacement.

Sample string : "The quick brown fox jumps over the lazy dog."

In the above string replace all the fox with cat.

Sample Output:

```
Original string: The quick brown fox jumps over the
lazy dog.
```

```
New String: The quick brown cat jumps over the lazy
dog.
```

6) Write a Java program to convert all the characters in a string to lowercase and Uppercase.

```
Original String: The Quick BroWn FoX!
String in lowercase: the quick brown fox|
String in uppercase: THE QUICK BROWN FOX!
```

7) Write a Java program to find longest Palindromic Substring within a string.

Sample Output:

```
The given string is: thequickbrownfoxxofnworbquickthe
The longest palindrome substring in the giv
en string is; brownfoxxofnworb
The length of the palindromic substring is: 16
```

8) Write a Java program to find the second most frequent character in a given string.

Sample Output:

```
The given string is: successes
The second most frequent char in the string is: c
```

9) Write a Java program to print all permutations of a given string

with repetition

Sample Output:

```
The given string is: PQR
The permuted strings are:
PPP
PPQ
PPR
...
RRP
RRQ
RRR
```

10) Write a Java program to check whether two strings are interliving of a given string. Assuming that the unique characters in both strings.

Sample Output:

```
The given string is: PMQNO
The interleaving strings are MNO and PQ
The given string is interleaving: true

The given string is: PNQMO
The interleaving strings are MNO and PQ
The given string is interleaving: false
```

11) Write a Java program to read a string,if the string begins with "red" or "black" return that color string, otherwise return the empty string.

Sample Output:

```
The given strings is: blacksea
The string begins with the color: black
```

12) Write a Java program to print after removing duplicates from a given string

Sample Output:

```
The given string is: w3resource
After removing duplicates characters the new string is:
w3resouc
```

13) Write a Java program to find first non repeating character in a string.

Sample Output:

```
The given string is: gibblegabbler
The first non repeated character in String is: i
```

14) Write a Java program to divide a string in n equal parts.

Sample Output:

```
The given string is: abcdefghijklmnopqrstuvwxy
The string divided into 5 parts and they are:

abcde
fghij
klmno
pqrst
uvwxy
```

15) Write a Java program to check if two given strings are rotations of each other.

Sample Output:

```
The given strings are: ABACD  and  CDABA

The concatination of 1st string twice is: ABACDABACD
The 2nd string CDABA  exists in the new string.
```

> Strings are rotations of each other
