1. Read an Employee data with idno, name and mobilenumber (regular expression) and compare the mobile number must have only 10 digits name can consists of only alphabets , space character idno number consists of 5 digits

```
package Tsgol.com;
import java.util.Scanner;
import java.util.regex.Pattern;
public class Employeevalidator {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
//Regular Expression Pattern
String mobilePattern = "\\d{10}"; // 10 digits
String namePattern = "[A-Za-z ]+"; // alphabets and space
characters
System.out.println("Enter Employee ID:");
String id = sc.next();
System.out.println("Enter Employee name:");
String name = sc.next();
System.out.println("Enter Employee Mobile number:");
String mobile = sc.next();
// Validating mobile number
if (!Pattern.matches(mobilePattern, mobile)) {
System.out.println("Invalid mobile number!");
return;
}
// Validating name
if (!Pattern.matches(namePattern, name)) {
System.out.println("Invalid name!");
return;
}
System.out.println("Employee data is valid!");
}
}
```

```
Output:
Enter Employee ID:
402
Enter Employee name:
Manasaveena
Enter Employee Mobile number:
9346409582
Employee data is valid!
```

2. Write a mutithreading program,
thread 1 : to display all perfect numbers,
thread 2 : to display factorial value of numbers from 1 to 10.

```
package Tsgol.com;
class PerfectNumberThread implements Runnable {
```

```
@Override
public void run() {
System.out.println("Perfect Numbers:");
for (int i = 1; i <= 1000; i++) {
try {
if (isPerfectNumber(i)) {
System.out.println(i);
}
} catch (Exception e) {
System.out.println("An exception occurred: " +
e.getMessage());
}
}
}
private boolean isPerfectNumber(int number) throws Exception {
if (number < 1) {
throw new Exception("Number must be greater than 0.");
}
int sum = 0;
for (int i = 1; i < number; i++) {
if (number % i == 0) {
sum += i;
}
}
return sum == number;
}
}
class FactorialThread implements Runnable {
@Override
public void run() {
//System.out.println("Factorial Values:");
for (int i = 1; i <= 10; i++) {
try {
Thread.sleep(2000);
System.out.println("Factorial value :");
System.out.println(i + "! = " + calculateFactorial(i));
} catch (Exception e) {
System.out.println("An exception occurred: " +
e.getMessage());
}
}
}
private int calculateFactorial(int number) throws Exception {
if (number < 0) {
throw new Exception("Number must be non-negative.");
}
if (number == 0) {
return 1;
}
int factorial = 1;
for (int i = 1; i <= number; i++) {
```

```
factorial *= i;
}
return factorial;
}
}
package Tsgol.com;
public class Multithreading {
     public static void main(String[] args) {
     Thread perfectNumberThread = new Thread(new
     PerfectNumberThread());
     Thread factorialThread = new Thread(new
FactorialThread());
     perfectNumberThread.start();
     factorialThread.start();
     }

}
```

Output:

```
Perfect Numbers:
6
28
496
Factorial value :
1! = 1
Factorial value :
2! = 2
Factorial value :
3! = 6
Factorial value :
4! = 24
Factorial value :
5! = 120
Factorial value :
6! = 720
Factorial value :
7! = 5040
Factorial value :
8! = 40320
Factorial value :
9! = 362880
Factorial value :
10! = 3628800
```

3. Write a program to read the data from file.

```
package Tsgol.com;
import java.io.*;
public class Readdata {
     public static void main(String[] args) throws IOException
```

```
      {
      FileReader fr=new FileReader("d:\\manasa\\Testfile.txt");
      BufferedReader br=new BufferedReader(fr);
      String str=null;
      while( true )
      {
      try
      {
      str=br.readLine(); // read from file
      if(str.equals(null))
      break;
      System.out.println(str);
      }
      catch(NullPointerException e)
      { break; }
      }
      br.close();
      fr.close();

}
}
```

```
Output:
Multithreading in java is a process of executing multiple
threads simultaneously.
```

4. Write a program to write the content to file in append mode.

```
package Tsgol.com;
import java.io.*;
public class Writedata {
      public static void main(String[] args) throws
IOException
      {
      DataInputStream dis = new DataInputStream(System.in);
      //FileWriter fw = new FileWriter("filename and
path",appendmode);
      FileWriter fw = new
FileWriter("d:\\manasa\\Testfile.txt",true);
      //Used to write data to file with the help of filewriter
object
      BufferedWriter br=new BufferedWriter(fw);
      String str=null;
      int size;
      while( true )
      {
      System.out.println("Enter file input");
      str=dis.readLine();
      if(str.equals("null"))
      break;
```

```
size=str.length();
br.write(str,0,size); //write to file
br.write("\n");
}
br.close();
fw.close();
}
}
```

Output:

Java Thread allows us to create a lightweight process that executes some tasks.