Q1. 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 regexp.com;

import java.util.Scanner;

import java.util.regex.Pattern;

public class Employee {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Regular expression patterns

String mobilePattern = "\\d{10}"; // 10 digits

String namePattern = "[A-Za-z ]+"; // alphabets and space
characters

System.out.println("Enter employee ID: ");

int id = sc.nextInt();

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;
}
// If both mobile number and name are valid
System.out.println("Employee data is valid!");
}
}
```

OUTPUT :

```
Enter employee ID:
102412
Enter employee Name:
Saikumar
Enter employee Mobile number:
8790835454
Employee data is valid!
```

Q2. Write a mutithreading program,

thread 1 : to display all perfect numbers,

thread 2 : to display factorial value of numbers from 1 to 10.

```
package multiprocess.com;
public class Fact_Perfect {
public static void main(String[] args) {
```

```
Thread perfectNumberThread = new Thread(new
PerfectNumberThread());

Thread factorialThread = new Thread(new FactorialThread());

perfectNumberThread.start();

factorialThread.start();

}

}

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(2500);

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;
}
}
```

Output :

```
Perfect Numbers:
6
28
496
Factorial Value:
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
```

```
8! = 40320

9! = 362880

10! = 3628800
```

Q3. Write a program to read the data from file.

```
package file.com;

import java.io.*;

public class Read_data {

public static void main(String[] sun) throws IOException

{

FileReader fr=new FileReader("d:\\sai\\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:

```
java is a robust language.
```

Q4. write a program to write the content to file in append mode.

```
package file.com;

import java.io.*;

public class Write_data {

public static void main(String[] args) throws IOException

{

DataInputStream dis = new DataInputStream(System.in);

//used to open the file for wr

//FileWriter fw = new FileWriter("filename and
path",appendmode);

FileWriter fw = new FileWriter("d:\\sai\\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:

```
Enter file input

java is a robust language.
```