Name : Anupam Kunwar
Reg : 19BCE1369

Week-9_Part-2

THREADS

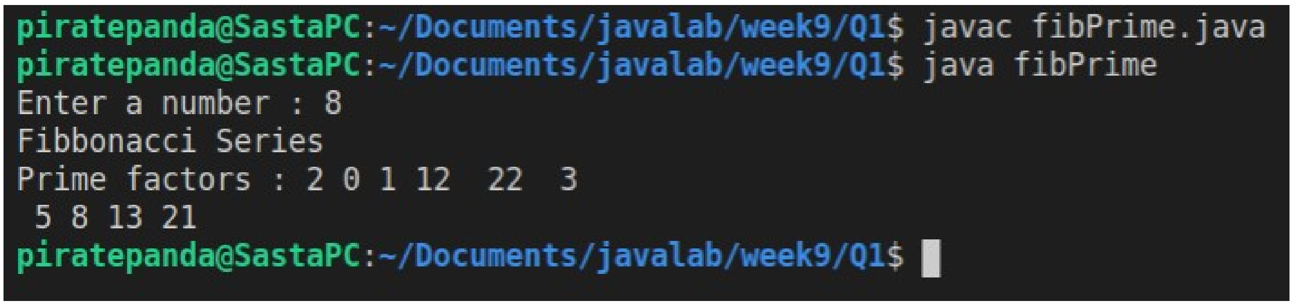
Q2.

Solution :
```
import java.util.Scanner;

class generators {
synchronized void generate(int n) {
int a, b, c, i;
a = 0;
b = 1;
i = 0;
System.out.println("Fibbonacci Series");
System.out.print(a + " " + b);
while (i <= n - 2) {
c = a + b;
System.out.print(" " + c);
a = b;
b = c;
i++;
try {
Thread.sleep(400);
} catch (Exception e) {
System.out.println(e);
}
}
System.out.println();
}

synchronized void findPrime(int n) {
System.out.print("Prime factors : ");
while (n % 2 == 0) {
System.out.print(2 + " ");
n /= 2;
try {
Thread.sleep(400);
} catch (Exception e) {
System.out.println(e);
}
}
for (int i = 3; i <= Math.sqrt(n); i += 2) {
while (n % i == 0) {
System.out.print(i + " ");
n /= i;
try {
Thread.sleep(400);
} catch (Exception e) {
System.out.println(e);
}
}
}
if (n > 2)
System.out.print(n);
System.out.println();
}

}

class fibbonacci extends Thread {
generators obj;
int n;

fibbonacci(generators obj, int n) {
this.obj = obj;
this.n = n;
}

public void run() {
obj.generate(n);
}
}

class Prime extends Thread {
int n;
generators obj;

Prime(generators obj, int a) {
n = a;
this.obj = obj;
}

public void run() {
obj.findPrime(n);
}
}

public class fibPrimeSync {

public static void main(String[] args) {
Scanner in = new Scanner(System.in);
int n;
System.out.print("Enter a number : ");
n = in.nextInt();
validate(n);
generators obj = new generators();
fibbonacci fib = new fibbonacci(obj, n);
Prime pri = new Prime(obj, n);
fib.start();
```

```
pri.start();
}
}
```

Output :

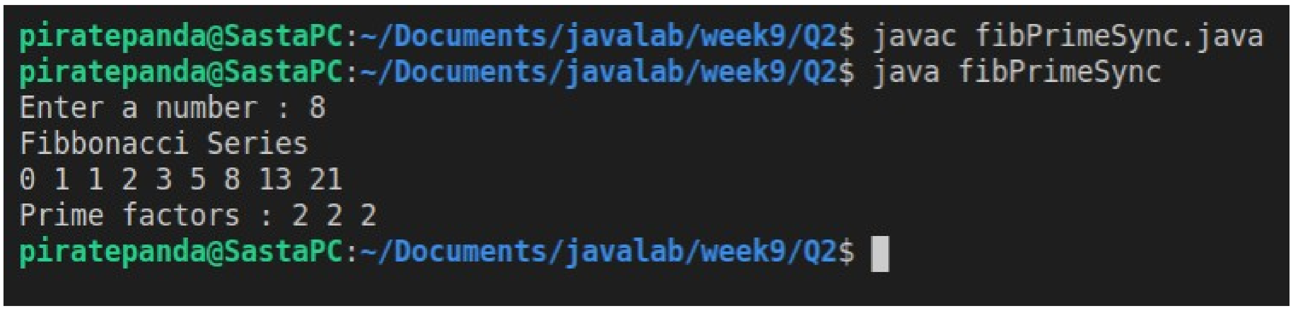
Without Synchronization :



```
piratepanda@SastaPC:~/Documents/javalab/week9/Q1$ javac fibPrime.java
piratepanda@SastaPC:~/Documents/javalab/week9/Q1$ java fibPrime
Enter a number : 8
Fibbonacci Series
Prime factors : 2 0 1 12  22  3
 5 8 13 21
piratepanda@SastaPC:~/Documents/javalab/week9/Q1$ 
```

.

With Synchronization :



```
piratepanda@SastaPC:~/Documents/javalab/week9/Q2$ javac fibPrimeSync.java
piratepanda@SastaPC:~/Documents/javalab/week9/Q2$ java fibPrimeSync
Enter a number : 8
Fibbonacci Series
0 1 1 2 3 5 8 13 21
Prime factors : 2 2 2
piratepanda@SastaPC:~/Documents/javalab/week9/Q2$ 
```

.

Q3.

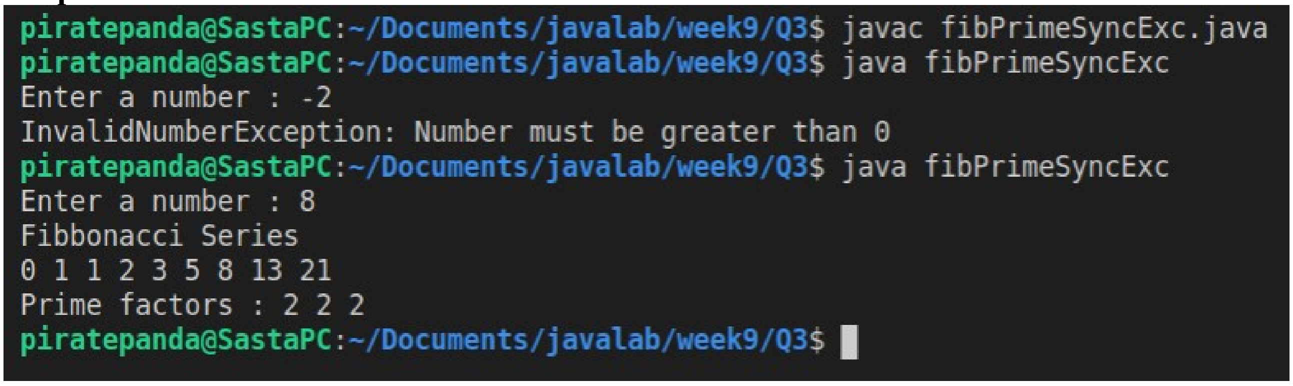
Sloution :

```
import java.util.Scanner;

class InvalidNumberException extends Exception {
InvalidNumberException(String s) {
super(s);
}
}

class generators {
synchronized void generate(int n) {
int a, b, c, i;
a = 0;
b = 1;
i = 0;
System.out.println("Fibbonacci Series");
System.out.print(a + " " + b);
while (i <= n - 2) {
```

```
c = a + b;
System.out.print(" " + c);
a = b;
b = c;
i++;
try {
Thread.sleep(400);
} catch (Exception e) {
System.out.println(e);
}
}
System.out.println();
}

synchronized void findPrime(int n) {
System.out.print("Prime factors : ");
while (n % 2 == 0) {
System.out.print(2 + " ");
n /= 2;
try {
Thread.sleep(400);
} catch (Exception e) {
System.out.println(e);
}
}
for (int i = 3; i <= Math.sqrt(n); i += 2) {
while (n % i == 0) {
System.out.print(i + " ");
n /= i;
try {
Thread.sleep(400);
} catch (Exception e) {
System.out.println(e);
}
}
}
if (n > 2)
System.out.print(n);
System.out.println();
}

}

class fibbonacci extends Thread {
generators obj;
int n;

fibbonacci(generators obj, int n) {
this.obj = obj;
this.n = n;
}
```

```
public void run() {
obj.generate(n);
}
}

class Prime extends Thread {
int n;
generators obj;

Prime(generators obj, int a) {
n = a;
this.obj = obj;
}

public void run() {
obj.findPrime(n);
}
}

public class fibPrimeSyncExc {

static void validate(int n) throws InvalidNumberException {
if (n < 0)
throw new InvalidNumberException("Number must be greater than 0");
}

public static void main(String[] args) {
Scanner in = new Scanner(System.in);
int n;
System.out.print("Enter a number : ");
n = in.nextInt();
try {
validate(n);
generators obj = new generators();
fibbonacci fib = new fibbonacci(obj, n);
Prime pri = new Prime(obj, n);
fib.start();
pri.start();
} catch (InvalidNumberException e) {
System.out.println(e);
}

}
}
```

Output :



```
piratepanda@SastaPC:~/Documents/javalab/week9/Q3$ javac fibPrimeSyncExc.java
piratepanda@SastaPC:~/Documents/javalab/week9/Q3$ java fibPrimeSyncExc
Enter a number : -2
InvalidNumberException: Number must be greater than 0
piratepanda@SastaPC:~/Documents/javalab/week9/Q3$ java fibPrimeSyncExc
Enter a number : 8
Fibbonacci Series
0 1 1 2 3 5 8 13 21
Prime factors : 2 2 2
piratepanda@SastaPC:~/Documents/javalab/week9/Q3$
```

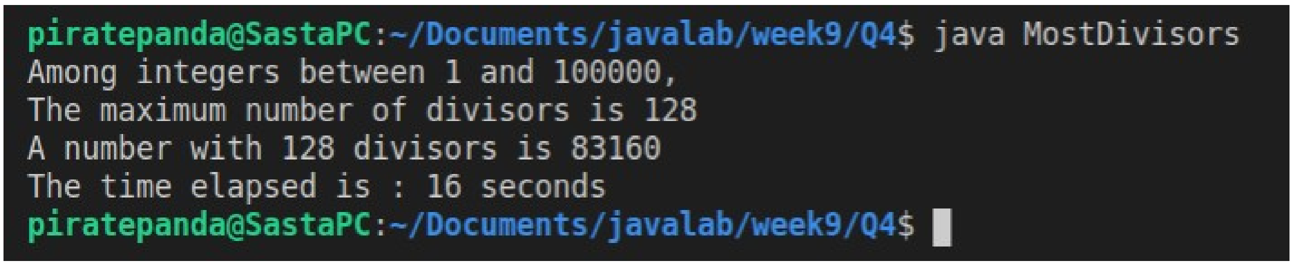
.

Q4.

Solution Without Threads :

```
import java.time.*;
public class MostDivisors {
public static void main(String[] args) {
Instant start = Instant.now();
int N;
int maxDivisors;
int numWithMax;
maxDivisors = 1;
numWithMax = 1;
for (N = 2; N <= 100000; N++) {
int D;
int divisorCount;
divisorCount = 0;
for (D = 1; D <= N; D++) {
if (N % D == 0)
divisorCount++;
}
if (divisorCount > maxDivisors) {
maxDivisors = divisorCount;
numWithMax = N;
}
}
System.out.println("Among integers between 1 and 100000,");
System.out.println("The maximum number of divisors is " + maxDivisors);
System.out.println("A number with " + maxDivisors + " divisors is " + numWithMax);
Instant end = Instant.now();
Duration timeElapsed = Duration.between(start,end);
System.out.println("The time elapsed is : "+timeElapsed.getSeconds()+" seconds");
}

}
```

Output :



```
piratepanda@SastaPC:~/Documents/javalab/week9/Q4$ java MostDivisors
Among integers between 1 and 100000,
The maximum number of divisors is 128
A number with 128 divisors is 83160
The time elapsed is : 16 seconds
piratepanda@SastaPC:~/Documents/javalab/week9/Q4$
```

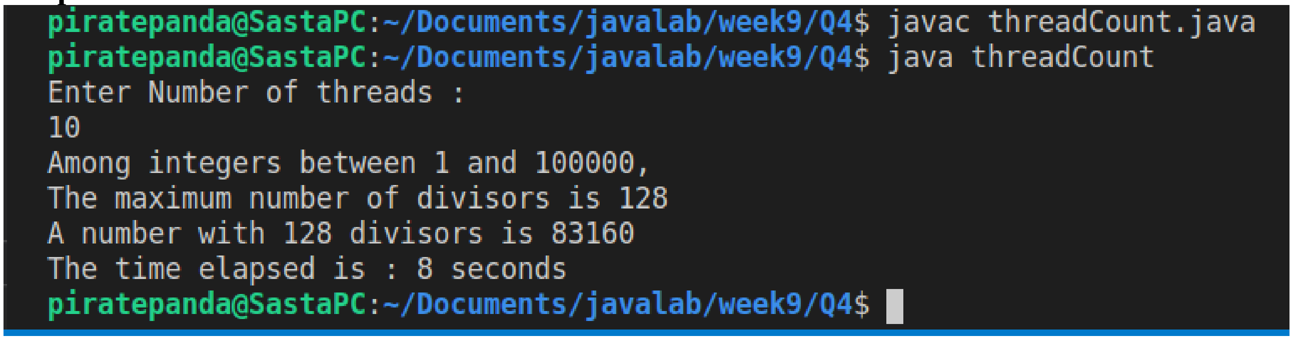
.

Solution with Threads :

```
import java.time.*;
import java.util.Scanner;

class counter extends Thread {
int start, end;

counter(int s, int e) {
start = s;
end = e;
}

public void run() {
threadCount obj = new threadCount();
int i, divisors;
for (i = start; i <= end; i++) {
divisors = countDivisors(i);
obj.report(divisors, i);
}
}

int countDivisors(int n) {
int i, divisors;
divisors = 0;
for (i = 1; i <= n; i++) {
if (n % i == 0)
divisors++;
}
return divisors;
}

}

public class threadCount {
static int maxDivisors = 1;
static int numWithMax = 1;

synchronized void report(int divisors, int i) {
if (divisors > maxDivisors) {
```

```
maxDivisors = divisors;
numWithMax = i;
}
}

public static void main(String[] args) {
Instant startTime = Instant.now();
Scanner in = new Scanner(System.in);
int N, i;
int numOfThread, intPerThread;
System.out.println("Enter Number of threads : ");
numOfThread = in.nextInt();
intPerThread = 100000 / numOfThread;
counter[] worker = new counter[numOfThread];
int start, end;
start = 1;
end = start + intPerThread - 1;
for (i = 0; i < numOfThread; i++) {
if (i == numOfThread - 1) {
end = 100000;
}
worker[i] = new counter(start, end);
start = end;
end = start + intPerThread - 1;
}
for (i = 0; i < numOfThread; i++)
worker[i].start();
for (i = 0; i < numOfThread; i++) {
while (worker[i].isAlive()) {
try {
worker[i].join();
} catch (InterruptedException e) {
}
}
}
System.out.println("Among integers between 1 and 100000,");
System.out.println("The maximum number of divisors is " + maxDivisors);
System.out.println("A number with " + maxDivisors + " divisors is " + numWithMax);
Instant endTime = Instant.now();
Duration timeElapsed = Duration.between(startTime, endTime);
System.out.println("The time elapsed is : " + timeElapsed.getSeconds() + " seconds");
}

}
```

Output :



```
piratepanda@SastaPC:~/Documents/javalab/week9/Q4$ javac threadCount.java
piratepanda@SastaPC:~/Documents/javalab/week9/Q4$ java threadCount
Enter Number of threads :
10
Among integers between 1 and 100000,
The maximum number of divisors is 128
A number with 128 divisors is 83160
The time elapsed is : 8 seconds
piratepanda@SastaPC:~/Documents/javalab/week9/Q4$
```