OSI Model

TCP/IP Model

จะเขียนโปรแกรมต้อง
บอกว่าใช้ อะไร

TCP/UDP

IP

OSI model   TCP/IP model

Arguments

```
public class TestJava
{
    public static void main(String[] args)
    {
        System.out.println("Number of argument : " + args.length);
        for(int i = 0; i < args.length; i++)
            System.out.println("Args[" + i + "] = " + args[i]);
    }
}
```

```
C:\WINDOWS\system32\cmd.exe

C:\>javac TestJava.java

C:\>java TestJava
Number of argument : 0

C:\>java TestJava Hello World 123
Number of argument : 3
Args[0] = Hello
Args[1] = World
Args[2] = 123
```

0 1 "2 3 AB C"

Type Conversion

☐ Arguments received from a command line are in the String format.

☐ So, if we want to use them as numbers, we need to convert them. We can use the static class below:

- **Integer.parseInt**(String   intValue)
- **Float.parseFloat(**String   floatValue)
- **Double.parseDouble**(String   doubleValue)

Example: Type Conversion

```java
import java.io.*;

public class TypeConversion {
    public static void main(String[] args) {
        String num1 = "1";
        int num2 = 2;
        System.out.println("Result1 = " + (num1 + num2));        12
        System.out.println("Result2 = " + (Integer.parseInt(num1)+num2));   3
    }
}
```

Quiz

```java
import java.io.*;

public class Exo1 {
    public static void main(String[] args) {
        if(args.length != 2) {
            System.out.println("Please enter 2 arguments");
            System.exit(1);
        }
        int num1 = Integer.parseInt(args[0]);
        int num2 = Integer.parseInt(args[1]);
        System.out.println("Result = " + (num1 + num2));
    }
}
```

- □ Find the output of this program when user runs it with the following commands:
    - □ java Exo1   please enter........
    - □ java Exo1   125   please enter.......
    - □ java Exo1   25   15   Result = 40
    - □ java Exo1   25   a   Exeption

Example: Class IOException

- □ java.lang.**Exception**   คลาสใหญ่สุด
    - □ java.io.**IOException**
        - ■ java.io.**CharConversionException**
        - ■ java.io.**EOFException**
        - ■ java.io.**FileNotFoundException**
        - ■ java.io.**InterruptedIOException**
        - ■ java.io.**ObjectStreamException**
            - ■ java.io.**InvalidClassException**
            - ■ java.io.**InvalidObjectException**
            - ■ java.io.**NotActiveException**
            - ■ java.io.**NotSerializableException**
            - ■ java.io.**OptionalDataException**
            - ■ java.io.**StreamCorruptedException**
            - ■ java.io.**WriteAbortedException**
        - ■ java.io.**SyncFailedException**
        - ■ java.io.**UnsupportedEncodingException**
        - ■ java.io.**UTFDataFormatException**

Example: Class Exception

- □ class java.lang.**Exception**
    - □ class java.lang.**ClassNotFoundException**
    - □ class java.lang.**CloneNotSupportedException**
    - □ class java.lang.**IllegalAccessException**
    - □ class java.lang.**InstantiationException**
    - □ class java.lang.**InterruptedException**
    - □ class java.lang.**NoSuchFieldException**
    - □ class java.lang.**NoSuchMethodException**
    - □ class java.lang.**RuntimeException**
        - ■ class java.lang.**ArithmeticException**
        - ■ class java.lang.**ArrayStoreException**
        - ■ class java.lang.**ClassCastException**
        - ■ class java.lang.**IllegalArgumentException**
            - ■ class java.lang.**IllegalThreadStateException**
            - ■ class java.lang.**NumberFormatException**
        - ■ class java.lang.**IllegalMonitorStateException**
        - ■ class java.lang.**IllegalStateException**
        - ■ class java.lang.**IndexOutOfBoundsException**
            - ■ class java.lang.**ArrayIndexOutOfBoundsException**
            - ■ class java.lang.**StringIndexOutOfBoundsException**
        - ■ class java.lang.**NegativeArraySizeException**
        - ■ class java.lang.**NullPointerException**
        - ■ class java.lang.**SecurityException**
        - ■ class java.lang.**UnsupportedOperationException**

Fixed the problem of "ArrayIndexOutofBoundsException"

```java
import java.io.*;

public class Exo1 {
    public static void main(String[] args) {
        try {
            int num = Integer.parseInt(args[0]);
            System.out.println("Result = " + num);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Usage : java Exo1 <number>");
        }
    }
}
```

- □ In your opinion, what would be the output of the following command:
  - □ java Exo1 Hello

Fixed the problem of "NumberFormatException"

```java
import java.io.*;

public class Exo1 {
    public static void main(String[] args) {
        try {
            int num = Integer.parseInt(args[0]);
            System.out.println("Result = " + num);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Usage : java Exo1 <number>");
        } catch (NumberFormatException ee) {
            System.out.println("Usage : java Exo1 <number>");
        }
    }
}
```

Make it easier to catch "Exception"

- □ The Exception class is the parent class of:
  - ▪ Class NumberFormatException
  - ▪ Class ArrayIndexOutOfBoundsException
- □ So, we can catch all exceptions by **catching the Exception class.**

```java
import java.io.*;

public class Exo1 {
    public static void main(String[] args) {
        try {
            int num = Integer.parseInt(args[0]);
            System.out.println("Result = " + num);
        } catch (Exception e) {
            System.out.println("Usage : java Exo1 <number>");
        }
    }
}
```

OutputStream

- □ The basic class for data transmission is

  java.io.OutputStream

- □ It has important methods: a[] = a b c d e
  - ▪ public abstract void **write(int b)** throws IOException
    write(a) -> a b c d e
  - ▪ public void **write(byte[ ] data)** throws IOException
    write(a,1,2) -> b c   จุดเริ่มต้น
  - ▪ public void **write(byte[ ] data, int offset, int length)** throws IOException
    ส่งทีเดียว
  - ▪ public void **flush( )** throws IOException
    flush อัตโนมัติ
  - ▪ public void **close( )** throws IOException

InputStream

- The basic class for data reception is

  java.io.InputStream

- It has important methods:

  b[] = _ _ _ _ _   abcdefg

  - public abstract int **read( )** throws IOException
    อ่านตามขนาดของ array
    read(b) => a b c d e
  - public int **read(byte[ ] input)** throws IOException
    read(b,1,2) => _ a b _ _
  - public int **read(byte[ ] input, int offset, int length)** throws IOException
  - public long **skip(long n)** throws IOException
  - public void **available( )** throws IOException
  - public void **close( )** throws IOException

Java and data file

- Managing files in Java can be done in various ways.
- The simplest method is to use the **File class.**
- **Example:**
  - File f = new File(String *filename*);
  - File f = new File(String *pathname*, String *filename*);
  - File f = new File(File *pathname*, String *filename*);

Java File Methods

- Details about each method of java **File** class can be found in the JavaDoc manual.
- The important methods are:
  - boolean **delete**();   delete a file/directory.
  - boolean **exists**();   check if a file/directory exists
  - boolean **isDirectory**();   check if it is a directory
  - boolean **isFile**();   check if it is a file
  - long   **length**();   get the size of a file/directory
  - File[] **listFiles**();   list file/directory name in that directory in an array of File type.
  - String[] **list**();   list file/directory name in that directory in an array of String type.
  - String **getName**();   get only file/directory name (removed path)

Example 1

```java
import java.io.*;

public class Example1 {
    public static void main(String[] args) {
        File f  = new File("myFile.txt");
        if(!f.exists()) {
            System.out.println("File does not exist");
            System.exit(1);
        }

        if(f.isFile()) {
            System.out.println("myFile.txt is a File");
            System.out.println("File size = " + f.length());
        } else if(f.isDirectory()) {
            System.out.println("myFile.txt is a directory");
        } else {
            System.out.println("....");
        }
    }
}
```

Example 2

```java
import java.io.*;

public class Example2 {
    public static void main(String[] args) {
        File f = new File("C:\\AppServ");
        File list[] = f.listFiles();

        for(int i = 0; i < list.length; i++) {
            if(list[i].isFile()) {
                System.out.println(list[i].getName() + " is a file");
            } else if (list[i].isDirectory()) {
                System.out.println(list[i].getName() + " is a directory");
            } else {
                System.out.println(list[i].getName() + " is unknown");
            }
        }
    }
}
```

```java
import java.io.*;

public class ReadFile {
    public static void main(String[] args) {
        try {
            int n;
            byte[] b = new byte[16];

            File f = new File("myfile.txt");
            FileInputStream fin = new FileInputStream(f);
            while((n = fin.read(b)) > 0) {
                String data = new String(b, 0, n);
                System.out.print(data);
            }

            fin.close();
        } catch(Exception e) { e.printStackTrace(); }
    }
}
```

Writing data to a file.

Example: Usage of PrintWriter

```java
import java.io.*;

public class WriteFile {
    public static void main(String[] args) {
        try {
            String msg = "Hello World";
            File f = new File("myfile.txt");
            FileOutputStream fout = new FileOutputStream(f);
            byte[] b = msg.getBytes();
            fout.write(b);
            fout.close();
        } catch(Exception e) { e.printStackTrace(); }
    }
}
```

```java
import java.io.*;

public class PwTest {
    public static void main(String[] args) {
        try {
            String msg = "Hello World";
            File f = new File("myfile.txt");
            FileOutputStream fout = new FileOutputStream(f);
            PrintWriter pout = new PrintWriter(fout);
            pout.print(msg);
            pout.flush();
            fout.close();
        } catch(Exception e) { e.printStackTrace(); }
    }
}
```

Example: Usage of BufferedReader (1)

```java
import java.io.*;

public class BrTest {
    public static void main(String[] args) {
        try {
            String msg;
            File f = new File("myfile.txt");
            FileInputStream fin  = new FileInputStream(f);
            InputStreamReader ir = new InputStreamReader(fin);
            BufferedReader br     = new BufferedReader(ir);

            while((msg = br.readLine()) != null)
                System.out.println(msg);

            fin.close();
        } catch(Exception e) { e.printStackTrace(); }

    }
}
```

Example: Usage of BufferedReader (2)

```java
import java.io.*;

public class BrTest2 {
    public static void main(String[] args) {
        try {
            String msg;
            File f = new File("myfile.txt");
            BufferedReader br     = new BufferedReader(
                              new InputStreamReader(
                              new FileInputStream(f)));

            while((msg = br.readLine()) != null)
                System.out.println(msg);

        } catch(Exception e) { e.printStackTrace(); }

    }
}
```

Example: Thread

```java
import java.io.*;

public class TwoThread extends Thread {
    public void run( ) {  ต้องมี
        for(int i = 0; i < 10; i++) {
            System.out.println("New Thread");
        }
    }

    public static void main(String[] args) {
        TwoThread tt = new TwoThread();
        tt.start();
                        สรุปคือทำงานพร้อมกัน
ทำงานเลย ไม่รอให้ รัน เสร็จ
        for(int i = 0; i < 10; i++) {
            System.out.println("Main Thread");
        }
    }
}
```

What do you think is the output of the program ?

Example: Sleep

```java
import java.io.*;

public class TestSleep {
    public static void main(String[] args) {
        System.out.println("Hello");

        try {
            Thread.sleep(3000);
        } catch (Exception e) {}

        System.out.println("Bye Bye");

    }
}
```

Example: multiple threads with sleep

```java
import java.io.*;

public class MultiThread extends Thread {
        String myName;
        long sleepTime;

        public MultiThread(String myName, long sleepTime) {
                this.myName   = myName;
                this.sleepTime = sleepTime;
        }

        public void run() {
                for(int i = 0; i < 5; i++) {
                        System.out.println(myName);
                        try {
                                Thread.sleep(sleepTime);
                        } catch(Exception e) {}
                }
        }

        public static void main(String[] args) {
                MultiThread t1 = new MultiThread("-1-", 1000);
                MultiThread t2 = new MultiThread("-2-", 2000);
                MultiThread t3 = new MultiThread("-3-", 3000);

                t1.start();
                t2.start();
                t3.start();
        }
}
```

4 Thread
t1 t2 t3 main

t1	t2	t3
-1-	-2-	-3-
5s	10s	15s

โปรแกรมนี้ทำงาน 15 s

ถ้ามี 1 thread ทำงานอยู่โปรแกรมจะยังไม่จบ

ผลการรัน

```
-1-
-3-
-2-
-1-
-2-
-1-
-3-
-1-
-2-
-1-
-3-
-2-
-2-
-3-
-3-
```

Example: Thread

```java
import java.io.*;

public class TwoThread implements Runnable {
        public void run( ) {
                for(int i = 0; i < 10; i++) {
                        System.out.println("New Thread");
                }
        }

        public static void main(String[] args) {
                TwoThread tt = new TwoThread();
                Thread t = new Thread(tt);
                t.start();

                for(int i = 0; i < 10; i++) {
                        System.out.println("Main Thread");
                }
        }
}
```

The differences of 2 methods

(1) Method: extends Thread

```java
import java.io.*;

public class TwoThread extends Thread {
        public void run( ) {
                for(int i = 0; i < 10; i++) {
                        System.out.println("New Thread");
                }
        }

        public static void main(String[] args) {
                TwoThread tt = new TwoThread();
                tt.start();

                for(int i = 0; i < 10; i++) {
                        System.out.println("Main Thr...");
                }
        }
}
```

Class modifier
(1) extends Thread
(2) implements Runnable

Creating and Invoking a Thread Object.

```java
import java.io.*;

public class TwoThread implements Runnable {
        public void run( ) {
                for(int i = 0; i < 10; i++) {
                        System.out.println("New Thread");
                }
        }

        public static void main(String[] args) {
                TwoThread tt = new TwoThread();
                Thread t = new Thread(tt);
                t.start();

                for(int i = 0; i < 10; i++) {
                        System.out.println("Main Thread");
                }
        }
}
```

(2) Method: implements Runnable

Example: summation program

```java
import java.io.*;

public class Sum {
        int from;
        int where;
        int result = 0;

        public Sum(int from, int where) {
                this.from  = from;
                this.where = where;
        }

        public void run() {
                for(int i = from; i <= where; i++) {
                        result += i;
                }
        }

        public int getResult() {
                return result;
        }

        public static void main(String[] args) {
                Sum s = new Sum(0, 1000000);
                s.run();
                System.out.println("Result = " + s.getResult());
        }
}
```

Thread issues

```java
import java.io.*;

public class SumThreadWrong implements Runnable {
    int from;
    int where;
    int result = 0;

    public SumThreadWrong(int from, int where) {
        this.from  = from;
        this.where = where;
    }

    public void run() {
        for(int i = from; i <= where; i++) {
            result += i;
        }
    }

    public int getResult() {
        return result;
    }
}
```

Problem about the addition operation between the output of Thread 1 and Thread 2

```java
public static void main(String[] args) {
    int s = 0;
    SumThreadWrong s1 = new SumThreadWrong(0, 499999);
    SumThreadWrong s2 = new SumThreadWrong(500000, 1000000);
    Thread t1 = new Thread(s1);
    Thread t2 = new Thread(s2);
    try {
        t1.start();  t2.start();
        s = s1.getResult() + s2.getResult();
    } catch(Exception e){}
    System.out.println("Result = " + s);
}
```

Usage: join( )

```java
import java.io.*;

public class SumThread implements Runnable {
    int from;
    int where;
    int result = 0;

    public SumThread(int from, int where) {
        this.from  = from;
        this.where = where;
    }

    public void run() {
        for(int i = from; i <= where; i++) {
            result += i;
        }
    }

    public int getResult() {
        return result;
    }
}
```

```java
public static void main(String[] args) {
    int s = 0;
    SumThread s1 = new SumThread(0, 499999);
    SumThread s2 = new SumThread(500000, 1000000);
    Thread t1 = new Thread(s1);
    Thread t2 = new Thread(s2);

    try {
        t1.start();  t2.start();
        t1.join();  t2.join();
        s = s1.getResult() + s2.getResult();
    } catch(Exception e){}
    System.out.println("Result = " + s);
}
```

DeadLock (1)

- Deadlock is a situation when more than 2 threads interlocks. ขัดกัน

```java
public class Friend {
    private String name;

    public Friend(String name) {
        this.name = name;
    }

    public String getName() {
        return this.name;
    }

    public synchronized void bow(Friend bower) {
        System.out.println(name + ": " + bower.getName() + " has bowed to me.");
        bower.bowBack(this);
    }

    public synchronized void bowBack(Friend bower) {
        System.out.println(name + ": " + bower.getName() + "has bowed back to me.");
    }

    public void run() {

    }
}
```
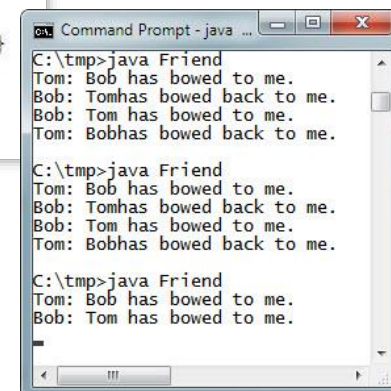
มีโอกาสที่จะ deadlock เพราะ มี synchronized ซ้อนกัน

Deadlock (2)

```java
public static void main(String[] args) {
    final Friend tom = new Friend("Tom");
    final Friend bob = new Friend("Bob");
    new Thread(new Runnable() {
        public void run() { tom.bow(bob);}
    }).start();
    new Thread(new Runnable() {
        public void run() { bob.bow(tom);}
    }).start();
}
```
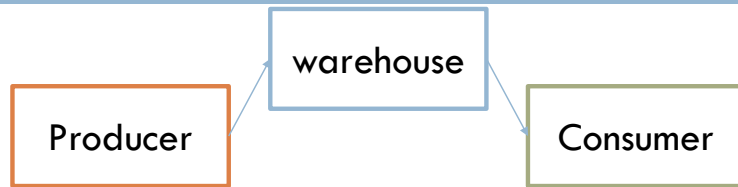
```
Command Prompt - java ...

C:\tmp>java Friend
Tom: Bob has bowed to me.
Bob: Tomhas bowed back to me.
Bob: Tom has bowed to me.
Tom: Bobhas bowed back to me.

C:\tmp>java Friend
Tom: Bob has bowed to me.
Bob: Tomhas bowed back to me.
Bob: Tom has bowed to me.
Tom: Bobhas bowed back to me.

C:\tmp>java Friend
Tom: Bob has bowed to me.
Bob: Tom has bowed to me.
```

Producer-Consumer Problem

- □ producer-consumer problem
  - ▣ Producer produces a product and stores it in a warehouse.
  - ▣ Consumer takes a product out of the warehouse.
  - ▣ Warehouse can store only 1 product.
  - ▣ Producer has to wait producing products if the warehouse is full.
  - ▣ Consumer has to wait taking products if the warehouse is empty.
- □ Time usage in producing or taking a product is a random number between 0 − 999 ms

Class : Warehouse (v.1)

```java
public class Warehouse {
    volatile int productID;
    volatile boolean empty = true;

    public synchronized void put(int productID) {
        while (!empty) { }   ไม่ควรอยู่ใน synchronized
        empty = false;
        this.productID = productID;
    }

    public synchronized int take() {
        while (empty) { }
        int result = this.productID;
        empty = true;
        return result;
    }
}
```

Class: Producer (v.1)

```java
import java.util.*;

public class Producer extends Thread {
    Warehouse w;

    public Producer(Warehouse w) {
        this.w = w;
    }

    public void run() {
        Random r = new Random();
        for(int i = 0; i < 10; i++) {
            int id = r.nextInt(100);   แรนดอม 0-99
            System.out.println("Producer: try to put product with id = " + id);
            w.put(id);
            System.out.println("Producer: put product with id = " + id);
            try {
                Thread.sleep(r.nextInt(1000));
            } catch(Exception e) {}
        }
    }
}
```

Class: Consumer (v.1)

```java
import java.util.*;

public class Consumer extends Thread {
    Warehouse w;

    public Consumer(Warehouse w) {
        this.w = w;
    }

    public void run() {
        Random r = new Random();
        for(int i = 0; i < 10; i++) {
            System.out.println("Consumer : try to take product");
            int id = w.take();
            System.out.println("Consumer : take product with id = " + id);
            try {
                Thread.sleep(r.nextInt(1000));
            } catch(Exception e){}
        }
    }
}
```
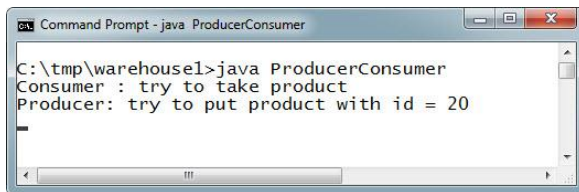
Class : ProducerConsumer (main)

```java
public class ProducerConsumer {
    public static void main(String[] args) {
        Warehouse w = new Warehouse();
        Producer p = new Producer(w);
        Consumer c = new Consumer(w);
        p.start(); c.start();
    }
}
```

```
Command Prompt - java ProducerConsumer

C:\tmp\warehouse1>java ProducerConsumer
Consumer : try to take product
Producer: try to put product with id = 20
```

Deadlock…??!!
Where is my
wrong code ??!

Fixed: Warehouse (v.2)

```java
public class Warehouse {
    volatile int productID;
    volatile boolean empty = true;

    public synchronized boolean put(int productID) {
        if (!empty) return false;
        empty = false;
        this.productID = productID;
        return true;
    }

    public synchronized int take() {
        if (empty) return -1;
        int result = this.productID;
        empty = true;
        return result;
    }
}
```

Fixed: Producer (v.2)

```java
import java.util.*;

public class Producer extends Thread {
    Warehouse w;

    public Producer(Warehouse w) {
        this.w = w;
    }

    public void run() {
        Random r = new Random();
        for(int i = 0; i < 10; i++) {
            int id = r.nextInt(100);
            System.out.println("Producer: try to put product with id = " + id);
            while(!w.put(id));
            System.out.println("Producer: put product with id = " + id);
            try {
                Thread.sleep(r.nextInt(1000));
            } catch(Exception e) {}
        }
    }
}
```
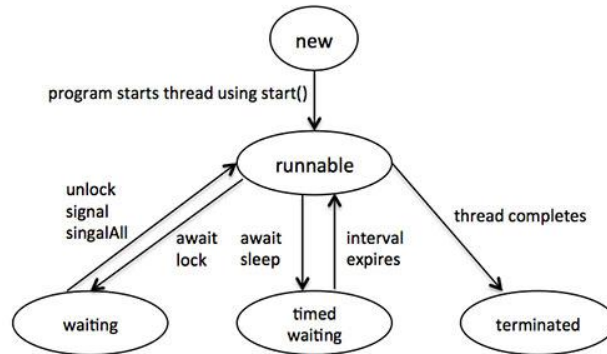
Fixed: Consumer (v.2)

```java
import java.util.*;

public class Consumer extends Thread {
    Warehouse w;

    public Consumer(Warehouse w) {
        this.w = w;
    }

    public void run() {
        int id;
        Random r = new Random();
        for(int i = 0; i < 10; i++) {
            System.out.println("Consumer : try to take product");
            while((id = w.take()) == -1);
            System.out.println("Consumer : take product with id = " + id);
            try {
                Thread.sleep(r.nextInt(1000));
            } catch(Exception e){}
        }
    }
}
```

What do you think
about this
program??

Wait/Notify/NotifyAll

- When a thread needs to wait some data from another thread, it can invoke the method **wait()** to wait the notification from another thread. หยุดรอ เหมือนหรับ ไม่ได้ใช้ซีพียู
- **notify()** is a method to send the notication to 1 thread (random) to wake it up from **wait().** เหมือนส่งไลน์ แต่สุ่ม ปลุกตัวเดียว
- **notifyAll()** ...is a method to send the notification to wake all of waiting threads up. ปลุกหมดเลย


```
                    ┌─────┐
                    │ new │
                    └─────┘
                        │
   program starts thread using start()
                        │
                        ▼
                  ┌──────────┐
      ┌──────────▶│ runnable │◀──────────┐
      │           └──────────┘           │
   unlock          │      │              │
   signal     await   await    interval  thread completes
   singalAll   lock   sleep    expires
      │           │      │         │        │
      ▼           ▼      ▼         ▼        ▼
  ┌─────────┐  ┌─────────┐   ┌────────────┐
  │ waiting │  │  timed  │   │ terminated │
  └─────────┘  │ waiting │   └────────────┘
               └─────────┘
```

Fixed: Warehouse (v.3)

Use Class Producer and Consumer v.1

```java
public class Warehouse {
    volatile int productID;
    volatile boolean empty = true;

    public synchronized void put(int productID) {
        if(!empty) {
            try {
                wait();
            } catch(Exception e) {}
        }
        this.productID = productID;
        empty = false;
        notify();   // มีเพื่อ ปลุก wait()
    }

    public synchronized int take() {
        if(empty) {
            try {
                wait();
            } catch(Exception e) {}
        }
        int result = this.productID;
        empty = true;
        notify();
        return result;
    }
}
```

Example: LinkedList Class

```java
import java.util.LinkedList;

public class LinkedListTest {
    public static void main(String[] args) {
        LinkedList<String> myList = new LinkedList();
        String m;

        myList.offer("1"); myList.offer("2"); myList.offer("3");
        myList.offer("4"); myList.offer("5"); myList.offer("6");

        System.out.println(myList);

        m = myList.poll();
        System.out.println("Output = " + m);
        m = myList.poll();
        System.out.println("Output = " + m);

        System.out.println(myList);
    }
}
```

```
run:
[1, 2, 3, 4, 5, 6]
Output = 1
Output = 2
[3, 4, 5, 6]
```
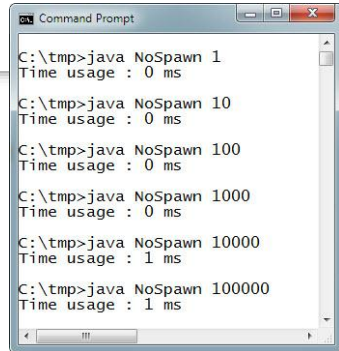
Thread issue – 2 (1)

```java
public class Spawn implements Runnable {
    public void run() { }

    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        long startTime = System.currentTimeMillis();
        for(int i = 0; i < n; i++) {
            Spawn s = new Spawn();
            Thread t = new Thread(s);
            t.start();
            try {
                t.join();
            } catch(Exception e) { }
        }
        long stopTime = System.currentTimeMillis();
        System.out.println("Time usage : " + (stopTime - startTime) + " ms");
    }
}
```
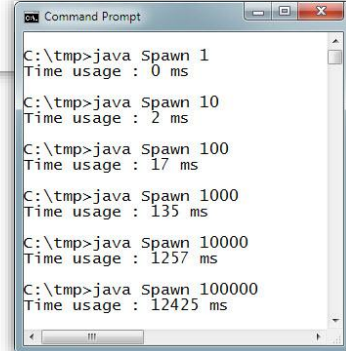จะได้เวลาว่าใช้ไปกี่millisec

```java
public class NoSpawn  {

    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        long startTime = System.currentTimeMillis();
        for(int i = 0; i < n; i++) { }
        long stopTime = System.currentTimeMillis();
        System.out.println("Time usage : " + (stopTime - startTime) + " ms");
    }
}
```

```
Command Prompt

C:\tmp>java NoSpawn 1
Time usage : 0 ms

C:\tmp>java NoSpawn 10
Time usage : 0 ms

C:\tmp>java NoSpawn 100
Time usage : 0 ms

C:\tmp>java NoSpawn 1000
Time usage : 0 ms

C:\tmp>java NoSpawn 10000
Time usage : 1 ms

C:\tmp>java NoSpawn 100000
Time usage : 1 ms
```

```
Command Prompt

C:\tmp>java Spawn 1
Time usage : 0 ms

C:\tmp>java Spawn 10
Time usage : 2 ms

C:\tmp>java Spawn 100
Time usage : 17 ms

C:\tmp>java Spawn 1000
Time usage : 135 ms

C:\tmp>java Spawn 10000
Time usage : 1257 ms

C:\tmp>java Spawn 100000
Time usage : 12425 ms
```

Sum.java — Source

```java
package sum;
public class Sum extends Thread {
    int from;
    int where;
    double result = 0;
    public Sum(int from, int where) {
        this.from = from;
        this.where = where;
    }
    public void run() {
        for (int i = from; i <= where; i++) {
            result += Math.log(a: Math.sqrt(a: i));
        }
    }
    public double getResult() {
        return result;
    }
    public static void main(String[] args) {
        Sum t1 = new Sum(from: 0, where: 250000000);
        Sum t2 = new Sum(from: 250000001, where: 500000000);
        Sum t3 = new Sum(from: 500000001, where: 750000000);
        Sum t4 = new Sum(from: 750000001, where: 1000000000);
        try {
            t1.start(); t1.join();
            t2.start(); t2.join();
            t3.start(); t3.join();
            t4.start(); t4.join();
            double r = t1.getResult() + t2.getResult();
            System.out.println("Result = " + r);
        } catch (Exception e) {}
    }
}
```

BadThread.java — Source

```java
package badthread;
public class BadThread {
    public static void main(String[] args) {
        BankAccount bankAcct = new BankAccount(money: 1000);
        BankBranch b1 = new BankBranch(bankAcct, method:"deposit", money: 10000);
        BankBranch b2 = new BankBranch(bankAcct, method:"withdraw", money: 10000);
        b1.start();
        b2.start();
        try{
            b1.join();
            b2.join();
        }catch(Exception e){}
        System.out.println("Balance = "+bankAcct.getBalance());
    }
}
```

BankAccount.java — Source

```java
package badthread;

public class BankAccount {
    volatile int money = 0;
    Object o = new Object();
    public BankAccount(int money) {
        this.money = money;
    }
    public void deposit(int money) {
        synchronized (o) {
            for (int i = 0; i < money; i++) {
                this.money++;
            }
        }
    }
    public void withdraw(int money) {
        synchronized (o) {
            for (int i = 0; i < money; i++) {
                this.money--;
            }
        }
    }
    public int getBalance() {
        return money;
    }
}
```

Sum.java — Source

```java
package sum;
import java.io.File;
import java.io.FileOutputStream;
import java.io.PrintWriter;
public class Sum extends Thread {
    int sum = 0;
    public void run() {
        try {
            File f = new File(pathname: "output.txt");
            FileOutputStream fout = new FileOutputStream(file: f);
            PrintWriter pout = new PrintWriter(out: fout);
            pout.print(i: sum);
            pout.flush(); fout.close();
        } catch (Exception e) {}
    }
    public Sum(int sum) {
        this.sum = sum;
    }
    public static void main(String[] args) {
        try {
            for (int i = 0; i < args.length; i++) {
                Integer.parseInt(args[i]);
            }
        } catch (NumberFormatException e) {
            System.out.println(x: "error args");
            System.exit(status: 0);
        }
        try {
            int total=0;
            for (int i = 0; i < args.length; i++) {
                total += Integer.parseInt(args[i]);
            }
            Sum s = new Sum(sum: total);
            s.start();
        } catch (Exception e) {}
    }
}
```

4. จงเขียนโปรแกรม Sum.java โดยโปรแกรมจะรับอาร์กิวเม็นต์กี่ตัวก็ได้ที่เป็นจำนวนเต็ม ซึ่งโปรแกรมจะคำนวณ
ผลรวมของทุกอาร์กิวเม็นต์และเขียนผลลัพธ์จากการคำนวณลงไปในไฟล์ชื่อ output.txt ถ้าอาร์กิวเม็นต์บางตัว
ไม่ใช่จำนวนเต็มจะให้โปรแกรมแสดงคำว่า "Error args" บนหน้าจอ และหยุดการทำงานทันที (12 คะแนน)

ตัวอย่างการใช้งานโปรแกรม:

คำสั่ง	ผลลัพธ์
java Sum 5 4 9 1	จะได้ไฟล์ output.txt ภายในจะมีเลข 19 (เกิดจาก 5+4+9+1)
java Sum	จะได้ไฟล์ output.txt ภายในจะมีเลข 0
java Sum 1 A 2	แสดงคำว่า Error args บนหน้าจอ ไม่เกิดไฟล์ output.txt

BankBranch.java — History

```java
package badthread;

public class BankBranch extends Thread {
    BankAccount bankAcct = null;
    String method = null;
    int money = 0;
    public BankBranch(BankAccount bankAcct, String method, int money) {
        this.bankAcct = bankAcct;
        this.method = method;
        this.money = money;
    }
    public void deposit(int money) {
        bankAcct.deposit(money);
    }
    public void withdraw(int money) {
        bankAcct.withdraw(money);
    }
    public void run() {
        if (method.equals(anObject: "deposit")) {
            deposit(money);
        } else {
            withdraw(money);
        }
    }
}
```

JavaTextCopy.java

```java
3   import java.io.BufferedReader;
4   import java.io.File;
5   import java.io.FileInputStream;
6   import java.io.FileOutputStream;
7   import java.io.InputStreamReader;
8   import java.io.PrintWriter;
9   public class JavaTextCopy {
10
11      public static void main(String[] args) {
12          if(args.length != 2){
13              System.out.println("Uasge: java javaTextCopy <source file> <destination file>");
14          }
15          else{
16              try{
17                  String msg;
18                  File data1 = new File(args[0]);
19                  FileInputStream fin = new FileInputStream(data1);
20                  InputStreamReader ir = new InputStreamReader(fin);
21                  BufferedReader br = new BufferedReader(ir);
22
23                  File data2 = new File(args[1]);
24                  FileOutputStream fout = new FileOutputStream(data2);
25                  PrintWriter pout = new PrintWriter(fout);
26                  while ((msg=br.readLine()) != null){
27                      pout.println(msg);
28                  }
29                  br.close();
30                  pout.close();
31              }catch (Exception e) {
32                  e.printStackTrace();
33              }
34          }
```

JavaThread.java

```java
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/1
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Ma
4   */
5   package javathread;
6
7   /**
8    *
9    * @author thasa
10   */
11  public class JavaThread extends Thread{
12      int n;
13      public JavaThread(int n){
14          this.n = n;
15      }
16      public void run(){
17          System.out.println(n + " " + "Hello Word");
18      }
19
20      public static void main(String[] args) {
21          if(args.length != 1){
22              System.out.println("Please enter one number.");
23          }else{
24              try{
25                  int c=Integer.parseInt(args[0]);
26                  for(int i=0;i<c;i++){
27                      JavaThread t = new JavaThread(i);
28                      t.start();
29                  }
30              }catch(NumberFormatException e){}
31          }
32      }
```

JavaTwoThread 1.java

```java
11  public class JavaTwoThread implements Runnable {
12
13      int from;
14      int to;
15      int sum = 0;
16
17      public JavaTwoThread(int from, int to) {
18          this.from = from;
19          this.to = to;
20      }
21
22      public void run() {
23          for (int i = from; i <= to; i++) {
24              sum += i;
25          }
26          try {
27              if (to == 5000) {
28                  Thread.sleep(5000);
29              } else {
30                  Thread.sleep(10000);
31              }
32          } catch (Exception e) {}
33      }
34
35      public int getSum() {
36          return sum;
37      }
38  }
39  public static void main(String[] args) {
40      JavaTwoThread j1 = new JavaTwoThread(1, 5000);
        JavaTwoThread j2 = new JavaTwoThread(5001, 10000);
        Thread t1 = new Thread(j1);
        Thread t2 = new Thread(j2);
        try {
            t1.start();
            t2.start();
            t1.join();
            t2.join();
        } catch (Exception e) {
        }
        System.out.println(j1.getSum() + j2.getSum());
    }
}
```

FileAVG.java

Source | History

```java
2   import java.io.BufferedReader;
3   import java.io.File;
4   import java.io.FileInputStream;
5   import java.io.FileOutputStream;
6   import java.io.InputStreamReader;
7   import java.io.PrintWriter;
8   public class FileAVG extends Thread {
9       float avg;
10      public void run() {
11          try {
12              File f = new File(pathname: "output.txt");
13              FileOutputStream fout = new FileOutputStream(file: f);
14              PrintWriter pout = new PrintWriter(out: fout);
15              pout.print("AVG = " +avg);
16              pout.flush();
17              pout.close();
18          } catch (Exception e) {}
19      }
20      public FileAVG(float avg) {
21          this.avg = avg;
22      }
23      public static void main(String[] args) {
24          try {
25              File f = new File(pathname: "C:\\Users\\thasa\\Desktop\\networkprogram\\NewFolder\\FileAVG\\input.txt");
26              BufferedReader br = new BufferedReader(
27                  new InputStreamReader(
28                      new FileInputStream(file: f)));
29              String msg;
30              float avg = 0;
31              while ((msg = br.readLine()) != null) {
32                  try {
33                      int n = Integer.parseInt(s: msg);
34                      avg += n;
35                  } catch (Exception e) {}
36              }
37              FileAVG a = new FileAVG(avg / 2);
38              a.start();
39          } catch (Exception e) {}
40      }
41  }
```

2. จงเขียนโปรแกรมภาษาจาวา FileAVG.java โดยโปรแกรมจะอ่านข้อมูลจากชื่อไฟล์ที่ป้อนเข้ามาจาก
อาร์กิวเมนต์ ซึ่งไฟล์ที่ป้อนมีตัวเลขที่เป็นจำนวนเต็ม 1 ค่าต่อ 1 บรรทัด จากนั้นโปรแกรมจะหาค่าเฉลี่ยของตัวเลข
ทั้งหมดเท่าที่อ่านได้ (เฉพาะตัวเลขจำนวนเต็มเท่านั้น) แล้วบันทึกค่าเฉลี่ยนั้นลงในไฟล์ลงนำค่าที่อ่านได้แต่ละบรรทัด
ไปรวมกับผลบวกเลขบรรทัด แล้วนำผลลัพธ์ที่ได้ไปเก็บไว้ในไฟล์ชื่อ c:\average.txt
สำหรับ Error ที่อาจเกิดจากการไม่ได้ป้อนอาร์กิวเมนต์ให้ดักจับตามความเหมาะสม (10 คะแนน)

ตัวอย่าง

java FileAVG c:\input.txt		
ตัวอย่างข้อมูลไฟล์ c:\input.txt	ไฟล์ c:\output.txt ที่โปรแกรมสร้าง	คำอธิบาย
3.5	AVG=12.5	ในไฟล์ input.txt มีเพียงแค่
4		บรรทัดที่มีเลข 4 และ 21 ที่
ABC		เป็นจำนวนเต็ม การหา
21		ค่าเฉลี่ยก็คือ (4 + 21) / 2 =
		12.5

JavaSyncTest.java

```java
11  public class JavaSyncTest implements Runnable {
12
13      static volatile int balance = 0;
14      static Object o = new Object();
15
16      public void run() {
17          for (int i = 0; i < 100000; i++) {
18              synchronized (o) {
19                  balance++;
20              }
21          }
22      }
23
24      public int getBalance(){
25          return balance;
26      }
27
28      public static void main(String[] args) {
29          JavaSyncTest j1 = new JavaSyncTest();
30          JavaSyncTest j2 = new JavaSyncTest();
31          JavaSyncTest j3 = new JavaSyncTest();
32
33          Thread t1 = new Thread(j1);
34          Thread t2 = new Thread(j2);
35          Thread t3 = new Thread(j3);
36
37          t1.start(); t2.start(); t3.start();
38          try {
39              t1.join();
40              t2.join();
41              t3.join();
42          } catch (Exception e) {}
43          System.out.println("Balance = "+ j1.getBalance());
```

Grade.java

```java
package grade;

public class Grade {

    public static void main(String[] args) {
        try {
            if (args.length != 1) {
                System.out.println("กรุณาป้อนคะแนนของคุณ");
            } else if (Integer.parseInt(args[0]) < 0 || Integer.parseInt(args[0]) > 100) {
                System.out.println("กรุณาป้อนตัวเลข 0-100");
            } else {
                int grade = Integer.parseInt(args[0]);
                if (grade >= 80) {
                    System.out.println("A");
                } else if (grade >= 70) {
                    System.out.println("B");
                } else if (grade >= 60) {
                    System.out.println("C");
                } else if (grade >= 50) {
                    System.out.println("D");
                } else {
                    System.out.println("F");
                }
            }

        } catch (NumberFormatException e) {
            System.out.println("กรุณาป้อนตัวเลขจำนวนเต็ม");
        }

    }
}
```

JavaList.java

```java
package javalist;

import java.io.File;
import java.rmi.server.ExportException;

public class JavaList {

    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Usage: java JavaList <File/Directory name>");
        } else {
            File f = new File(args[0]);
            if (f.exists()) {
                if (f.isFile()) {
                    System.out.println("File size = " + f.length());
                } else if (f.isDirectory()) {
                    String[] ff = f.list();
                    for (int i = 0; i < ff.length; i++) {
                        System.out.println(ff[i]);
                    }
                } else {
                    System.out.println("BUG");
                }
            } else {
                System.out.println("File not found");
            }
        }
    }
}
```

TestArgs.java

```java
package testargs;
public class TestArgs {
    public static void main(String[] args) {
        try{
            if(args.length != 2){
                System.out.println("Usage: java TestArgs <number1> <number2>");
            }else{
                float num1 = Float.parseFloat(args[0]);
                float num2 = Float.parseFloat(args[1]);
                System.out.println("number = "+ (num1*num2) );
            }
        }catch (NumberFormatException e){
            System.out.println("Please enter your float.");
        }
    }
}
```

JavaBinaryCopy.java

```java
package javabinarycopy;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
public class JavaBinaryCopy {
    public static void main(String[] args) {
        if (args.length != 2) {
            System.out.println("Usage: java JavaBinaryCopy <source file> <destination file>");
        } else {
            try {
                File data = new File(args[0]);
                FileInputStream fin = new FileInputStream(data);
                byte[] a = new byte[16];

                File f = new File(args[1]);
                FileOutputStream fout = new FileOutputStream(f);

                int n;
                while ((n = fin.read(a)) > 0) {
                    fout.write(a, 0, n);
                }
                fin.close();
                fout.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

Exam3.java

```java
package exam3;
public class Exam3 extends Thread {
    String s;
    long sleep;
    int count;
    public void run() {
        try {
            for (int i = 0; i < count; i++) {
                System.out.println(s);
            }
            Thread.sleep(sleep);
        } catch (Exception e) {}
    }

    public Exam3(String s, int sleep, int count) {
        this.s = s;
        this.sleep = sleep;
        this.count = count;
    }

    public static void main(String[] args) {
        if (args.length != 1) {
            System.out.println("Error argument");
            System.exit(0);
        }

        int n = Integer.parseInt(args[0]);
        for (int i = 1; i <= n; i++) {
            Exam3 e = new Exam3("Hello World", i * 1000, i);
            e.start();
//            try {
//                e.join();
//            } catch (Exception ee) {
//            }
        }
    }
}
```

3. จงเขียนโปรแกรมภาษาจาวาในแฟ้มข้อมูลชื่อ Exam3.java ซึ่ง Class Exam3 จะเป็น Class ที่สามารถทำงานเป็น Thread ด้วยวิธี extends Thread **(15 คะแนน)**

- โปรแกรมจะรับอาร์กิวเม็นต์ 1 ค่า คือ จำนวน thread ที่ต้องการสร้าง โดยจะส่ง thread ที่ถูกสร้างขึ้นนี้ จะพิมพ์ข้อความว่า "Hello World" เป็นจำนวนครั้งตามลำดับที่ถูกสร้างขึ้นมา พร้อมทั้งจะหลับหลังจาก พิมพ์ข้อความเสร็จเป็นจำนวน "วินาที" ตามลำดับที่ถูกสร้างขึ้นมา
- ถ้าผู้ใช้ป้อนจำนวนอาร์กิวเม็นต์ไม่เท่ากับ 1 ค่า หรือค่าที่ป้อนเข้ามาไม่ใช่จำนวนเต็ม ให้แสดงคำว่า "Error argument" แล้วจบโปรแกรม

<u>การใช้งานโปรแกรม:</u> ถ้าผู้ใช้ป้อน argument เป็น 3 ให้กับโปรแกรม โปรแกรม (main) จะสร้าง thread ขึ้นมา 3 threads โดย thread ตัวแรกจะพิมพ์คำว่า "Hello World" 1 ครั้ง และหลับไป 1 วินาที, thread ตัวที่ 2 จะ พิมพ์คำว่า "Hello World" 2 ครั้ง แล้วหลับไป 2 วินาที ส่วน thread ตัวสุดท้ายคือตัวที่ 3 จะพิมพ์คำว่า "Hello World" 3 ครั้ง แล้วหลับไป 3 วินาที ซึ่งการทำงานรวมของโปรแกรมจะใช้เวลาประมาณ 3 วินาที