

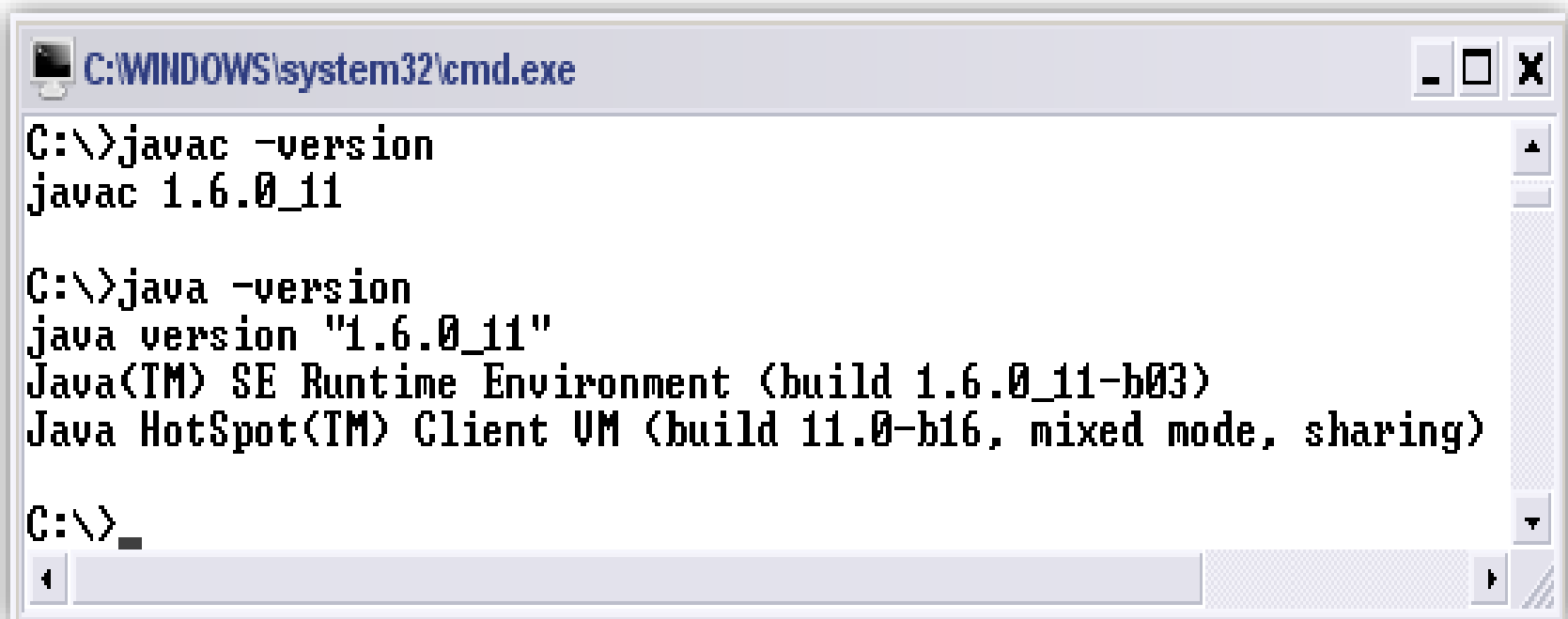
# OVERVIEW JAVA

030523313 - Network programming  
Asst. Prof. Dr. Choopan Rattanapoka

# Introduction

- Java จะคู่กับ motto “write once run anywhere”
- การพัฒนาโปรแกรมด้วยภาษา Java จะต้องติดตั้ง JDK ซึ่งจะประกอบด้วย
  - ▣ **javac** (java compiler) เป็นตัว compiler ที่แปลง source code (.java) ให้เป็น java bytecode (.class)
  - ▣ **java bytecode** เปรียบเสมือน executable file ที่สามารถนำไปใช้งานได้กับทุก OS ที่ support java
  - ▣ **java** เป็นตัว interpreter ที่จะอ่าน java bytecode เป็นภาษาเครื่อง

# ตรวจสอบความพร้อมของเครื่องที่ใช้พัฒนา Java



A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The command prompt displays the following text:

```
C:\>javac -version  
javac 1.6.0_11  
  
C:\>java -version  
java version "1.6.0_11"  
Java(TM) SE Runtime Environment (build 1.6.0_11-b03)  
Java HotSpot(TM) Client VM (build 11.0-b16, mixed mode, sharing)  
  
C:\>_
```

The window has a standard Windows interface with minimize, maximize, and close buttons in the top right corner. A scrollbar is visible on the right side of the text area.

# โครงสร้างภาษา Java

```
1  import java.net.*;
2  import java.io.*;
3
4  public class MyClass
5  {
6      public static void main(String[] args)
7      {
8          System.out.println("Hello World!");
9      }
10 }
```

- ❑ บรรทัด 1 - 2 : ส่วนของ import สำหรับเรียกใช้ class ที่มีอยู่ใน package นั้นๆ
- ❑ บรรทัด 4 : การประกาศชื่อ class จะต้องมียี่ห้อเหมือนกับชื่อ file ดังนั้น file นี้ต้องชื่อว่า MyClass.java
- ❑ บรรทัด 6 - 9 : เป็นส่วน main ของ java โดย java จะเริ่มทำงานที่บรรทัดนี้

# Java Package ที่จำเป็น

- ในการเขียนโปรแกรมภาษา **Java** ให้ทำงานเกี่ยวกับ **network** จะต้อง **import 2 packages** ที่จำเป็นคือ
  - ▣ **java.io** (บรรจุ Class ที่ทำงานเกี่ยวกับ **input/output**)
  - ▣ **java.net** (บรรจุ Class เกี่ยวกับการทำงานกับระบบเครือข่าย)
- ดังนั้นควรมี 2 บรรทัดนี้ขึ้นต้นใน **source code**

```
import    java.io.*;  
import    java.net.*;
```

# Java API

- เนื่องจาก **java** มี **class** และ **method** ใช้ให้อย่างมากมาย ทำให้ไม่สามารถจำได้หมด
- ในการพัฒนาโปรแกรมด้วยภาษา **Java** นั้นควรดู **API** จากเว็บ  
<http://java.sun.com/javase/6/docs/api/>  
(สำหรับ **java version 1.6.X**) ควบคู่ไปกับการพัฒนาโปรแกรม

# Java Command Line

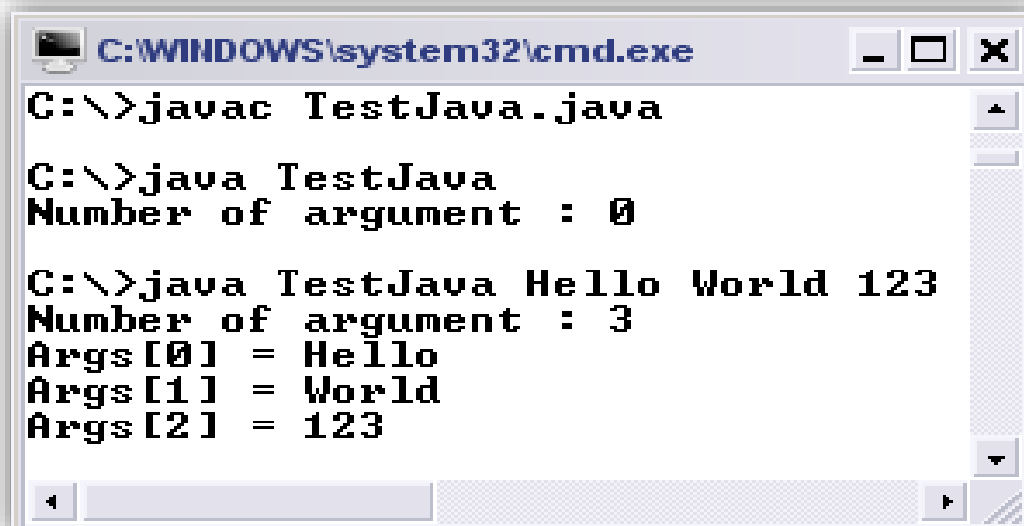
- ในโปรแกรมภาษา **Java** จะเหมือนกับภาษา **C** คือจะมีการทำงานเริ่มต้นที่ **function main**
- **function main** จะต้องอยู่ในรูปแบบต่อไปนี้เท่านั้น

**public static void main( String[ ] args )**

- ▣ โดย **args** ที่เป็น **parameter** ของ **function main** จะเป็นตัวรับค่า **arguments** จาก **command line** ซึ่งจะอยู่ในรูป **array** ของ **string**

# Arguments

```
1 public class TestJava
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Number of argument : " + args.length);
6         for(int i = 0; i < args.length; i++)
7             System.out.println("Args[" + i + "] = " + args[i]);
8     }
9 }
```



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
```

# Type Conversion

- เนื่องจากการรับ **argument** จาก **command line** นั้นจะอยู่ในรูป **String**
- ดังนั้นถ้าอยากรับค่าที่เป็นตัวเลขเพื่อใช้ในการคำนวณจะต้องแปลง **String** เป็น **data type** ที่เป็นตัวเลข สามารถเรียกใช้จาก **static class** ต่างๆ ต่อไปนี้เช่น
  - ▣ **Integer.parseInt(String intValue)**
  - ▣ **Float.parseFloat(String floatValue)**
  - ▣ **Double.parseDouble(String doubleValue)**

# ตัวอย่าง: Type Conversion

```
import java.io.*;

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

# แบบฝึกหัด

```
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));
    }
}
```

□ จงหาผลการรัน ถ้าผู้ใช้เรียกใช้งานด้วยคำสั่ง

□ java Exo1

□ java Exo1 125

□ java Exo1 25 15

□ java Exo1 25 a

# Java try-catch

- ในภาษา **Java** เมื่อมีการเรียกใช้งาน **method** และเกิดข้อผิดพลาดขึ้น (**error**) จะมีเหตุการณ์ที่เรียกว่า **Exception** เกิดขึ้น
- ใน **Class** มาตรฐานของ **Java** แต่ละ **method** จะมีการโยน (**throw**) **Exception** ออกจาก **method** เพื่อให้ผู้เรียกใช้ **method** สามารถจัดการเอง
- การเขียนโปรแกรมเพื่อดัก **Exception** ที่เกิดขึ้นใน **method** จะใช้ คำสั่ง **try** ตามด้วย { } และให้นำชุดคำสั่งที่ต้องการจะดักจับ **Exception** ไว้ข้างใน { }
- เมื่อมี **Exception** เกิดขึ้นโปรแกรมจะกระโดดไปยังส่วนของ **catch**

# ตัวอย่าง Class IOException

- java.lang.Exception
  - ▣ 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

# ตัวอย่าง 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](#)

# ตัวอย่างการดู API ใน Java

## □ Class Integer

- public static int **parseInt**(String s) throws NumberFormatException

Parses the string argument as a signed decimal integer. The characters in the string must all be decimal digits, except that the first character may be an ASCII minus sign '-' ('\u002D') to indicate a negative value.

- **Parameters:**s - a String containing the int representation to be parsed
- **Returns:**the integer value represented by the argument in decimal.
- **Throws:** NumberFormatException - if the string does not contain a parsable integer.

# ตัวอย่างโปรแกรมที่ไม่มีการดัก Exception

```
import java.io.*;

public class Exo1 {
    public static void main(String[] args) {
        int num = Integer.parseInt(args[0]);
        System.out.println("Result = " + num);
    }
}
```

- คิดว่าผลการรันจะเป็นเช่นไรถ้าผู้ใช้เรียกใช้งานด้วยคำสั่ง
  - java Exo1

# แก้ปัญหา ArrayIndexOutOfBoundsException

```
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>");
        }
    }
}
```

- คิดว่าผลการรันจะเป็นเช่นไรถ้าผู้ใช้เรียกใช้งานด้วยคำสั่ง
  - java Exo1 Hello

# แก้ปัญหา NumberFormatException

```
import java.io.*;

public class Ex01 {
    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 Ex01 <number>");
        } catch (NumberFormatException ee) {
            System.out.println("Usage : java Ex01 <number>");
        }
    }
}
```

# วิธีเขียนดักที่ Class Exception

- เนื่องจาก Class Exception เป็น Class แม่ของ
  - ▣ Class NumberFormatException
  - ▣ Class ArrayIndexOutOfBoundsException
- ดังนั้นการดักที่ Class แม่อย่างเดียวจะทำให้ดักได้หมด

```
import java.io.*;

public class Ex01 {
    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 Ex01 <number>");
        }
    }
}
```