

# Fundamental Data Types

ชนิดของข้อมูลแบบพื้นฐาน

# This chapter

- Primitive Types
- การประกาศค่าคงที่ในโปรแกรม (constant declaration)
- ตัวกระทำทางคณิตศาสตร์ (arithmetic operators)
- การเรียกใช้ static method
- สตริง

# Primitive Types

- byte (1-byte integer), short (2-byte integer), int (4-byte integer), long (8-byte integer)
- float (4-byte floating point number), double (8-byte floating point number)
- char (2-byte Unicode)
- boolean (1-bit truth value)

# คลาสสำหรับ big number ใน java.math

- Class BigInteger
- Class BigDecimal : floating point number ที่ไม่มี roundoff error
- ต้องใช้เมทอดสำหรับตัวกระทำ เช่น เมทอด multiply

# ค่าคงที่ (constants)

ตั้งชื่อให้ค่าคงที่ให้สื่อความหมาย  
ทำให้โปรแกรมอ่านง่าย

- จำนวนเต็ม เช่น 31
- จำนวนจริง เช่น 3.14159 -3.1E-4
- ตัวอักขระ เช่น '&'
- สายอักขระ เช่น "What is it ?"
- ตั้งชื่อให้ค่าคงที่ คล้ายชื่อตัวแปร โดยใช้คำว่า **final**

```
final float BPF = 212.0;  
public class Math { ...  
    public static final double E = 2.71028184590452354;  
    public static final double PI = 3.1415926535897923846;  
... }
```

ใช้ BPF, Math.E, Math.PI

# ฟังก์ชันทางคณิตศาสตร์ที่ไม่เป็นตัวกระทำในจาวา

ตัวอย่าง

- ยกกำลัง ใช้ `Math.pow(x, n)`
- รากที่สอง ใช้ `Math.sqrt(x)`

$$\text{คำนวณ } x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

>> ลองเขียนดู

ระวัง: ในภาษาจาวา `int/int` เป็น integer division ที่ปัดเศษทิ้ง เช่น `3/2` ได้ `1`

# การเปลี่ยนชนิด (Type casting) สำหรับจำนวน

*(typeName) expression*

```
double score = 92.1;  
final int FULL = 110;  
int percentRounded = (int) (score/FULL); // narrow casting
```

- Widening Casting
  - byte -> short -> char -> int -> long -> float -> double
  - จาวาทำให้อัตโนมัติ บางที เรียกว่า type coercion
- Narrowing casting (lossy conversion)
  - double -> float -> long -> int -> char -> short -> byte
  - โปรแกรมเมอร์ต้องระบุ

# การเรียกใช้ Static method

- บางที่เรียก class method
- เมื่อสร้างเมทอดชนิดนี้ในคลาส ต้องระบุที่ header ว่า static เช่น

```
public final class Math {  
    ...  
    public static double sqrt(double a) {...}  
    ...  
}
```
- จัดกลุ่มเมทอดไว้ในคลาสเดียวกัน เช่น class Math
- เรียกใช้ static method (หรือ class method) ได้โดยไม่ต้องอ้างถึง object เช่น `Math.sqrt(100)`



# Roundoff error

ถ้ากำหนด `double f = 4.35;`

- `(int)(100*f)` ได้ 434

ทำไม???

- 4.35 เป็นเลขฐานสอง

100.0101 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001  $1 \times 2^0$

- เมื่อเก็บในคอมพิวเตอร์ จะปรับเป็น

1.0001 0110 0110 0110 0110 0110 0110 0110 0110 0110 0110 0110 0110 011  $\times 2^2$

- 11 ใน 2 หลักสุดท้ายถูกตัดทิ้ง
- `100*f` ได้ 434.9999999999994

# ตัวอักขระ (Characters)

# ASCII code

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_ 0	<u>NUL</u> 0000	<u>SOH</u> 0001	<u>STX</u> 0002	<u>ETX</u> 0003	<u>EOT</u> 0004	<u>ENQ</u> 0005	<u>ACK</u> 0006	<u>BEL</u> 0007	<u>BS</u> 0008	<u>HT</u> 0009	<u>LF</u> 000A	<u>VT</u> 000B	<u>FF</u> 000C	<u>CR</u> 000D	<u>SO</u> 000E	<u>SI</u> 000F
1_ 16	<u>DLE</u> 0010	<u>DC1</u> 0011	<u>DC2</u> 0012	<u>DC3</u> 0013	<u>DC4</u> 0014	<u>NAK</u> 0015	<u>SYN</u> 0016	<u>ETB</u> 0017	<u>CAN</u> 0018	<u>EM</u> 0019	<u>SUB</u> 001A	<u>ESC</u> 001B	<u>FS</u> 001C	<u>GS</u> 001D	<u>RS</u> 001E	<u>US</u> 001F
2_ 32	<u>SP</u> 0020	<u>!</u> 0021	<u>"</u> 0022	<u>#</u> 0023	<u>\$</u> 0024	<u>%</u> 0025	<u>&amp;</u> 0026	<u>'</u> 0027	<u>(</u> 0028	<u>)</u> 0029	<u>*</u> 002A	<u>±</u> 002B	<u>·</u> 002C	<u>=</u> 002D	<u>˙</u> 002E	<u>/</u> 002F
3_ 48	<u>0</u> 0030	<u>1</u> 0031	<u>2</u> 0032	<u>3</u> 0033	<u>4</u> 0034	<u>5</u> 0035	<u>6</u> 0036	<u>7</u> 0037	<u>8</u> 0038	<u>9</u> 0039	<u>:</u> 003A	<u>:</u> 003B	<u>≤</u> 003C	<u>≡</u> 003D	<u>≥</u> 003E	<u>?</u> 003F
4_ 64	<u>@</u> 0040	<u>A</u> 0041	<u>B</u> 0042	<u>C</u> 0043	<u>D</u> 0044	<u>E</u> 0045	<u>F</u> 0046	<u>G</u> 0047	<u>H</u> 0048	<u>I</u> 0049	<u>J</u> 004A	<u>K</u> 004B	<u>L</u> 004C	<u>M</u> 004D	<u>N</u> 004E	<u>O</u> 004F
5_ 80	<u>P</u> 0050	<u>Q</u> 0051	<u>R</u> 0052	<u>S</u> 0053	<u>T</u> 0054	<u>U</u> 0055	<u>V</u> 0056	<u>W</u> 0057	<u>X</u> 0058	<u>Y</u> 0059	<u>Z</u> 005A	<u>[</u> 005B	<u>\</u> 005C	<u>]</u> 005D	<u>^</u> 005E	<u>_</u> 005F
6_ 96	<u>`</u> 0060	<u>a</u> 0061	<u>b</u> 0062	<u>c</u> 0063	<u>d</u> 0064	<u>e</u> 0065	<u>f</u> 0066	<u>g</u> 0067	<u>h</u> 0068	<u>i</u> 0069	<u>j</u> 006A	<u>k</u> 006B	<u>l</u> 006C	<u>m</u> 006D	<u>n</u> 006E	<u>o</u> 006F
7_ 112	<u>p</u> 0070	<u>q</u> 0071	<u>r</u> 0072	<u>s</u> 0073	<u>t</u> 0074	<u>u</u> 0075	<u>v</u> 0076	<u>w</u> 0077	<u>x</u> 0078	<u>y</u> 0079	<u>z</u> 007A	<u>{</u> 007B	<u> </u> 007C	<u>}</u> 007D	<u>~</u> 007E	<u>DEL</u> 007F

# Unicode

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+0E0x		<u>ก</u>	<u>ข</u>	<u>ฃ</u>	<u>ค</u>	<u>ศ</u>	<u>ฆ</u>	<u>ง</u>	<u>จ</u>	<u>ฉ</u>	<u>ช</u>	<u>ซ</u>	<u>ฌ</u>	<u>ญ</u>	<u>ฎ</u>	<u>ฏ</u>
U+0E1x	<u>ฐ</u>	<u>ฑ</u>	<u>ฒ</u>	<u>ณ</u>	<u>ด</u>	<u>ต</u>	<u>ถ</u>	<u>ท</u>	<u>ธ</u>	<u>น</u>	<u>บ</u>	<u>ป</u>	<u>ผ</u>	<u>ฝ</u>	<u>พ</u>	<u>ฟ</u>
U+0E2x	<u>ภ</u>	<u>ม</u>	<u>ย</u>	<u>ร</u>	<u>ฤ</u>	<u>ฌ</u>	<u>ภ</u>	<u>ว</u>	<u>ศ</u>	<u>ษ</u>	<u>ส</u>	<u>ห</u>	<u>ฬ</u>	<u>อ</u>	<u>ฮ</u>	<u>ย</u>
U+0E3x	<u>๕๕</u>	<u>๕๖</u>	<u>๕๗</u>	<u>๕๘</u>	<u>๕๙</u>	<u>๕๐</u>	<u>๕๑</u>	<u>๕๒</u>	<u>๕๓</u>	<u>๕๔</u>	<u>๕๕</u>					<u>฿</u>
U+0E4x	<u>๕๕</u>	<u>๕๖</u>	<u>๕๗</u>	<u>๕๘</u>	<u>๕๙</u>	<u>๕๐</u>	<u>๕๑</u>	<u>๕๒</u>	<u>๕๓</u>	<u>๕๔</u>	<u>๕๕</u>	<u>๕๖</u>	<u>๕๗</u>	<u>๕๘</u>	<u>๕๙</u>	<u>๕๐</u>
U+0E5x	<u>๕๕</u>	<u>๕๖</u>	<u>๕๗</u>	<u>๕๘</u>	<u>๕๙</u>	<u>๕๐</u>	<u>๕๑</u>	<u>๕๒</u>	<u>๕๓</u>	<u>๕๔</u>	<u>๕๕</u>	<u>๕๖</u>				
U+0E6x																
U+0E7x																

# Escape sequence

- ใช้จัดการกับ unprintable character
- ใช้ \ นำหน้า character (ใน character) เช่น
  - '\t' หมายถึง ตัวอักขระ tab
  - '\n' หมายถึง ตัวอักขระ newline
- ใช้ \ นำหน้า unicode เช่น
  - '\u0064' หมายถึง ตัวอักขระ d
  - '\u0E01' หมายถึง ตัวอักขระ ก

# แปลง character เป็นจำนวนเต็ม

- ใช้ type casting เพื่อแปลง character เป็นจำนวนเต็มไม่ได้

```
char code = '6';  
int num = (int)code; // Result: num = 54
```

- ค่าที่ได้เป็น ASCII code ของตัวอักขระ 6
- นั่นคือ ถ้าทำ type casting จาก character เป็น integer จะได้ ASCII code ของ character นั้น เช่น

```
char code = 'A';  
int num = (int)code; // num = 65
```

- ค่าที่ได้เป็น ASCII code ของตัวอักขระ A

# แปลงจำนวนเต็มเป็น character

- ใช้ type casting เพื่อแปลงจำนวนเต็มเป็น character ไม่ได้

```
int num = 54;  
char code = (char) num; // Result: code = '6'
```

- ค่าที่ได้เป็นตัวอักขระ 6 ที่แทนด้วย ASCII code 54
- นั่นคือ ถ้าทำ type casting จาก integer เป็น character จะได้ character ที่แทนด้วย ASCII code ที่มีค่านั้น เช่น

```
int num = 65;  
char code = (char) num; // Result: code = 'A'
```

- ค่าที่ได้เป็นตัวอักขระ A ที่แทนด้วย ASCII code 65

# คลาส String



# สายอักขระหรือสตริง (String)

- ในภาษาจาวา สตริงไม่เป็น primitive type แต่เป็น object ในคลาส String
- แต่ใช้ string literal เช่น "Java string" เป็น object ในคลาส String ได้
- "" เป็นสตริงว่าง (empty string)
- ประกาศตัวแปรที่เป็นวัตถุในคลาส String ได้ดังนี้  
`String address;`  
`String stName = "Tom";`

# เมทอด `length()`

- ความยาวของสตริง คือ จำนวน character ในสตริง
- สตริงว่างมีความยาว 0
- `"What is this ?"` มีความยาว 14
- `length()` เป็นเมทอดในคลาส `string` ที่ทำงานกับ object (สังเกตเทียบกับเมทอด `sqrt` ในคลาส `Math` )
- `stName.length()` หาความยาวของสตริงในตัวแปร `stName`
- `"What is this ?".length()` หาความยาวของสตริง `"What is this ?"`
- `"".length()` คืนค่า 0

# การต่อสตริง (string concatenation)

- ใช้ + เป็นตัวกระทำสำหรับต่อสตริงด้วย
- "Tom"+"@chula.ac.th" ได้ "Tom@chula.ac.th"
- ถ้าใช้ + กับสตริงและจำนวนเต็ม จะแปลงจำนวนเต็มเป็นสตริงแล้วนำมาต่อกัน เช่น

"age = "+1 ได้ "age = 1"

int x=1; String ch=x+"1"+1; ได้ ch เป็น "111"

int x=1; String ch=x+1+"1"; ได้ ch เป็น "21"

# แปลงสตริงเป็นจำนวน

- ใช้ type casting เพื่อแปลงสตริงเป็นจำนวนไม่ได้

```
String facultyCode = "23";  
int facultyNum = (int) facultyCode; // error
```

Error: String cannot be converted to int

- ใช้เมทอด Integer.parseInt, Double.parseDouble

```
String facultyCode = "23";  
int facultyNum = Integer.parseInt(facultyCode); //wrapper Integer
```

```
Scanner in = new Scanner(System.in);  
String price = in.nextLine();  
double vat = Double.parseDouble(price)*0.07; // wrapper Double
```

# แปลงจำนวนเป็นสตริง

- ใช้ type casting เพื่อแปลงจำนวนเป็นสตริงไม่ได้

```
int facultyNum = 23;  
String facultyCode = (String)facultyNum; //error
```

Error: int cannot be converted to String

- ใช้เมทอด `String.valueOf()`

```
int facultyNum = 23;  
String facultyCode = String.valueOf(facultyNum); // get "23"
```

```
double vat = 0.07;  
String vatStr = String.valueOf(vat); //get "0.07"
```

# เมทอด substring

- Substring เป็นเมทอดในคลาส string
- ใช้ *str.substring(startpoint, cutpoint)* คั้นค่าสตริงย่อยของ *str* เริ่มที่ตำแหน่ง *startpoint* ไปถึงตำแหน่ง *cutpoint-1*

```
String name="Timothy";  
int base=3;  
name.substring(0,base) ได้ผลลัพธ์เป็น "Tim"
```

# Escape sequence ในสตริง

- ใช้จัดการกับ unprintable character หรือ character ที่ใส่ไว้ใน "... " ไม่ได้
- ใช้ \ นำหน้า character คล้ายกับที่ใช้กับ character เช่น
  - "\" หมายถึง สตริงที่มีตัวอักขระ "
  - "\\" หมายถึง สตริงที่มีตัวอักขระ \
  - "C:\\Temp\\Doc\\data.txt" คือสตริง "C:\Temp\Doc\data.txt"

# Methods ในคลาส Scanner



# การอ่าน input ตามชนิด

- Scanner เป็นคลาสของจาวาที่สร้างไว้สำหรับการรับ input อยู่ใน package java.util

```
Scanner inp = new Scanner(System.in);
```

- สร้าง Scanner ที่รับ input จากคีย์บอร์ด โดยมี inp เป็น reference variable ของ Scanner นี้
- หลังจากนั้น จะเรียกเมทอดที่รับ input จากคีย์บอร์ดผ่านตัวแปรนี้ เช่น
  - `inp.nextLine()` อ่าน input เป็นสตริงจนเจอ enter
  - `inp.nextInt()` อ่าน input เป็นสตริงแล้วแปลงเป็นจำนวนเต็ม
  - `inp.nextDouble()` อ่าน input เป็นสตริงแล้วแปลงเป็นจำนวนจริง
- ถ้าอ่านมาแล้วแปลงเป็นชนิดที่กำหนดไม่ได้ จะเกิด error

# การอ่าน input ตามชนิด

```
import java.util.scanner
Scanner inp = new Scanner(System.in);
System.out.println(input.nextInt());
System.out.println(input.nextDouble());
System.out.println(input.nextLine());
```

- ถ้าผู้ใช้พิมพ์

```
9 32.10 this is a test<enter>
```

- ผลลัพธ์ที่แสดงออกมา คือ

```
9
```

```
32.1
```

```
this is a test
```