

## Lecture 3

Sunday, 5 January 2020 2:40 PM

Data types .Primitive (Stacks) ✓

-128 to 127.

Non-primitive (Heap) ✓

- ① byte → 1 byte →  $-2^7$  to  $2^7-1$
- ② short → 2 byte →  $-2^{15}$  to  $2^{15}-1$ .
- ③ int → 4 byte →  $-2^{31}$  to  $2^{31}-1$ .
- ④ long → 8 byte →  $-2^{63}$  to  $2^{63}-1$ .
- ⑤ float → 4 —
- ⑥ double → 8 —
- ⑦ boolean → 1 bit
- ⑧ char → 2 bytes.

$$1024 \rightarrow 1000$$

$$(2^{10}) \rightarrow$$

$$2^{10} \cdot 2^{10} \cdot 2^{10} \Rightarrow 2^{30}$$

$$\Rightarrow 1000 \times 1000 \times 1000$$

$$= 10^9.$$

int a = 10; literal.

byte b = 10;

int, short, byte, long. } int.

L.H.S = R.H.S.

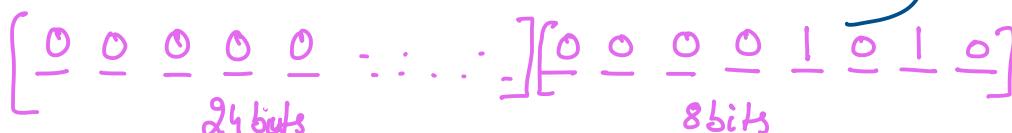
int a = 10; ✓

byte b = 10; [-128 to 127]

byte b = (byte)(10);

implicit typecast.

byte b = 10;



X

byte b = 130; // compile time error.

[-----][0000010] (X)

byte b = (byte)(130); Sys0(b) ✓  
 explicit type cast.

[10000010] → 130x

byte b = 10; 8bit ✓  
 short s = 20; 16bit ✓  
 int i = 30; 32bit ✓  
 long l = 40; 64bit ✓

l = b ✓  
 l = s ✓  
 l = i ✓

b = (byte)(s) 16bits

Part 1.

b = s x  
 b = i x  
 b = l x



s = b; ✓  
 s = i; x  
 s = l; x



i = b ✓  
 i = s ✓  
 i = l x

Part 2.

byte b = 10; ✓  
 Sys0(b); // 10.

b = 128; x  
 Sys0(b);  
 b = (byte)(128);  
 Sys0(b); // -128.

Part 3.

float f = 5.5 (double) x; ✓

float f = 5.5f; // explicit  
 double d = 5.5; ✓

4 byte → 8 byte x.

Part 4

```

    i = f;
    syso(i)
    f = i; // 5
    syso(f) // 5.0 ✓
  
```

char → 2 bytes.

Ascii

|            |           |          |
|------------|-----------|----------|
| 32 → space | 97 → 'a'  | 48 → '0' |
| 65 → 'A'   | 98        | 49 → '1' |
| 66 → 'B'   | 1         | 50 → '2' |
| 1          | 1         | 51 → '3' |
| 90 → 'Z'   | 122 → 'z' | 57 → '9' |

1 byte →

|   |   |   |   |   |   |   |   |               |
|---|---|---|---|---|---|---|---|---------------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |               |
|   |   |   |   |   |   |   |   | ↪ 256 numbers |

256 characters.

0-255 } Ascii.

2 bytes → 16 bits → 64000 nearly ↪ UNICODE.

256 → 0-255 → Ascii

256  
1  
1  
64000

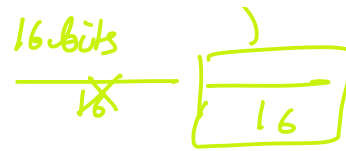
char ch = 'a'; ↪ 97 ↪  
syso(ch); // a

char ch = 99; 0-64000  
syso(ch) // c.

\*\*\* ch = 69000 % ? ↪ explicitly.

ch = (char)(69000); 32 bits

sys0(ch);



ch = 97;

ch = (ch + 2);

sys0(ch);

char + int  $\Rightarrow$  int // ch  $\Rightarrow$  97.  
64000

short sh = 10;  
ch = ch + sh;

2 byte + 4 byte  
(2 byte) . 4 byte

2 byte  
+ 2 byte

ch = 12;  
10 + 2;

how things are stored in Memory?.

+32  $\rightarrow$  Convert it to binary.



-ve 1 1 1 1 1 1 1 1  
M.S.B.  
+ve 0 1 1 1 1 1 1 1

256 numbers

$\hookrightarrow$  -127 to 127  $\rightarrow$

-1 to 1  $\rightarrow$  3

-2 to 2  $\rightarrow$  5

-3 to 3  $\rightarrow$  7

!

-127 to 127  $\rightarrow$  255

+0  
-0 } X

2's Complement

32 +ve Number

(1)  $\rightarrow$  Convert to binary

(2)  $\rightarrow$  store as it is.

0 0 1 0 0 0 0 0

-32 -ve number

- ① Convert to binary
- ② Take 1's Comp.
- ③ — 2's —.

$$\begin{array}{r}
 \text{①} \quad 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \\
 \text{②} \quad 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 \\
 \text{③} \quad \quad \quad \quad \quad \quad \quad + \ 1 \\
 \hline
 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \Rightarrow -32
 \end{array}$$

Read  $\rightarrow$   $\underline{0} \ \underline{0} \ \underline{1} \ \underline{0} \ \underline{0} \ \underline{0} \ \underline{0} \ \underline{0} \Rightarrow 32$   
 M.S.B  $\rightarrow$

$\rightarrow$   $\underline{1} \ \underline{1} \ \underline{1} \ \underline{0} \ \underline{0} \ \underline{0} \ \underline{0} \ \underline{0}$   
 M.S.B  $\rightarrow$

1's Comp  $\rightarrow$   $\underline{0} \ \underline{0} \ \underline{0} \ \underline{1} \ \underline{1} \ \underline{1} \ \underline{1} \ \underline{1}$   
 2's  $\rightarrow$   $\quad \quad \quad \quad \quad \quad \quad + \ 1$   
 $\underline{0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0} \rightarrow \text{decimal} \Rightarrow 32$

128 (byte)  
 byte b = 128;  
~~00000000~~  
 24 bits  
 M.S.B  
 Read  $\rightarrow$

byte b = 128;

-32

$\underline{1} \ \underline{0} \ \underline{0} \ \underline{0} \ \underline{0} \ \underline{0} \ \underline{0} \ \underline{0}$   
 8 bits

$$\begin{array}{r}
 0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \\
 \quad \quad \quad \quad \quad \quad \quad + \ 1 \\
 \hline
 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \Rightarrow -128
 \end{array}$$

F to C  $\Rightarrow (5/9) * (f - 32)$

Output

0  $-17.666 \dots$

20  $-6$

40  $4$

60  $15$

80  $26$

$f_{\min} = 0$

$f_{\max} = 300$

step = 20

$\boxed{(5) * (f - 32) / 9} x$



```

public static HelloFun() {
    // -> Syso("HelloFun");
    HelloGun();
}

```

H.F  
H.G  
H.G

```

void HelloGun() {
    Syso("Hello Gun");
}

```

int a;  
Syso(a); } x.  
psvm(String[] args) {  
 // Command line args.

Main  
 int p = 1000;  
 int r = 10;  
 int t = 1  
 int ans = si(p, r, t);  
 Syso(ans);  
 Syso(si(p, r, t));

10 20

int a = 10;

Stack  
 ans = 100  
 t = 1  
 r = 10  
 p = 1000  
 X.  
 100.

```

}
// arguments.
// local variables of si;
int si(int p, int r, int t) {
    int si = (p * r + t) / 100;
    return si;
}

```

```

psvm(String[] args) {

```

int a = 10;

int b = 20;

Syso(a + " " + b);

swap(a, b);

Syso(a + " " + b);

10 20 ✓  
 10 20 ✓  
 20 10 ✓  
 20 10 <-> 10.  
 X

```

Main {
    b = 20
    a = 10
}

void swap(int a, int b) {
    syso(a + " " + b);
    int temp = a;
    a = b;
    b = temp;
    syso(a + " " + b);
}

```

Q  $\Rightarrow$  K rotate  $\rightarrow$  32165 % 10, K

$K = K + \text{nod};$  5

$K = -3$

$K = -3 + 5$

$-4 + 5 = 1$

$K = 72 \% 5$   
 $= 2$

$\checkmark K=1 \rightarrow 53216$   $\checkmark$   $K=-1 \rightarrow 21653$   
 $\checkmark K=2 \rightarrow 65321$   $\checkmark$   $K=-2 \rightarrow 16532$   $\checkmark$   
 $K=3 \rightarrow 16532$   $\checkmark$   $K=-3 \rightarrow 65321$   $\checkmark$   
 $K=4 \rightarrow 21653$   $K=-4 \rightarrow 53216$   $\checkmark$   
 $K=5 \rightarrow 32165$   $K=-5 \rightarrow 32165$   
 $K=6 \rightarrow 53216$

-2 is same as 3  $\checkmark$

① Count digits  $\checkmark$  5.

②  $K = K \% \text{nod};$

③ if  $K < 0$ ,  $K = K + \text{nod};$

④  $\text{divisor} = \text{Math.pow}(10, K);$  100

65;

321

$\text{rem} = \text{num} \% \text{divisor}$

$\text{num} = \text{num} / \text{divisor}$

$\text{ans} = \text{rem} * \text{Math.pow}(10, \text{S}-2) + \text{num};$

$65 * 10^{5-2} + 321$

$65 * 10^{5-2} = 65000$   
 $+ 321$   


---

 $65321$