Princeton University
COS 217: Introduction to Programming Systems
C Primitive Data Types

Type: int

Description: A (positive or negative) integer.

Size: System dependent. Usually either 2 or 4 bytes.

Example Variable Declarations:

```
int iFirst;
int iSecond, iThird;
signed int iFourth;
```

Example Literals (assuming size is 4 bytes):

C Literal	Binary Representation	Note
123	00000000 00000000 00000000 01111011	decimal form
-123	11111111 11111111 11111111 10000101	negative form
2147483647	01111111 11111111 11111111 11111111	largest
-2147483648	10000000 00000000 00000000 00000000	smallest
0173	00000000 00000000 00000000 01111011	octal form
0x7B	00000000 00000000 00000000 01111011	hexadecimal form

Type: unsigned int

Description: A non-negative integer.

Size: System dependent. Usually either 2 or 4 bytes. sizeof(unsigned int) == sizeof(int).

Example Variable Declarations:

```
unsigned int uiFirst;
unsigned int uiSecond, uiThird;
```

Example Literals (assuming size is 4 bytes):

C Literal	Binary Representation	Note
123U	00000000 00000000 00000000 01111011	decimal form
4294967295U	11111111 11111111 11111111 11111111	largest
0U	00000000 00000000 00000000 00000000	smallest
0173U	00000000 00000000 00000000 01111011	octal form
0x7BU	00000000 00000000 00000000 01111011	hexadecimal form

Type: long

Description: A (positive or negative) integer.

Size: System dependent. Usually 4 bytes. sizeof(long) >= sizeof(int).

Example Variable Declarations:

```
long lFirst;
long lSecond, lThird;
long int lFourth;
signed long lFifth;
```

```
        signed long int lSixth;
```

Example Literals (assuming size is 4 bytes):

```
    C Literal        Binary Representation                       Note

    123L             00000000 00000000 00000000 01111011         decimal form
    -123L            11111111 11111111 11111111 10000101         negative form
    2147483647L      01111111 11111111 11111111 11111111         largest
    -2147483648L     10000000 00000000 00000000 00000000         smallest
    0173L            00000000 00000000 00000000 01111011         octal form
    0x7BL            00000000 00000000 00000000 01111011         hexadecimal form
```

Type: unsigned long

Description: A non-negative integer.

Size: System dependent. Usually 4 bytes. sizeof(unsigned long) == sizeof(long).

Example Variable Declarations:

```
    unsigned long ulFirst;
    unsigned long ulSecond, ulThird;
    unsigned long int ulFourth;
```

Example Literals (assuming size is 4 bytes):

```
    C Literal        Binary Representation                       Note

    123UL            00000000 00000000 00000000 01111011         decimal form
    4294967295UL     11111111 11111111 11111111 11111111         largest
    0UL              00000000 00000000 00000000 00000000         smallest
    0173UL           00000000 00000000 00000000 01111011         octal form
    0x7BUL           00000000 00000000 00000000 01111011         hexadecimal form
```

Type: char

Description: A (positive or negative) integer. Usually represents a character according to a
character code (e.g., ASCII).

Size: 1 byte.

Example Variable Declarations:

```
    char cFirst;
    char cSecond, cThird;
    signed char cFourth;
```

Example Literals (assuming the ASCII code is used):

```
    C Literal      Binary Representation      Note

    'a'            01100001                   character form
    (char)97       01100001                   decimal form
    (char)0141     01100001                   octal form
    (char)0x61     01100001                   hexadecimal form
    '\o141'        01100001                   octal character form
    '\x61'         01100001                   hexadecimal character form

    (char)123      01111011                   decimal form
    (char)-123     10000101                   negative form
    (char)127      01111111                   largest
    (char)-128     10000000                   smallest

    '\0'           00000000                   the null character
```

```
'\a'            00000111                    bell
'\b'            00001000                    backspace
'\f'            00001100                    formfeed
'\n'            00001010                    newline (system dependent)
'\r'            00001101                    carriage return
'\t'            00001001                    horizontal tab
'\v'            00001011                    vertical tab
'\\'            01011100                    backslash
'\''            00100111                    single quote
```

Type: unsigned char

Description: A non-negative integer. Usually represents a character according to a character code (e.g., ASCII).

Size: 1 byte.

Example Variable Declarations:

```
    unsigned char ucFirst;
    unsigned char ucSecond, ucThird;
```

Example Literals (assuming the ASCII code is used):

```
    C Literal                Binary Representation         Note

    (unsigned char)'a'       01100001                      character form
    (unsigned char)97        01100001                      decimal form
    (unsigned char)255       11111111                      largest
    (unsigned char)0         00000000                      smallest
```

Note: On most systems, "char" is the same as "signed char".
       On some systems, "char" is the same as "unsigned char".

Type: short

Description: A (positive or negative) integer.

Size: System dependent. Usually 2 bytes. sizeof(short) <= sizeof(int).

Example Variable Declarations:

```
    short sFirst;
    short sSecond, sThird;
    short int sFourth;
    signed short sFifth;
    signed short int sSixth;
```

Example Literals (assuming size is 2 bytes):

```
    C Literal                Binary Representation         Note

    (short)123               00000000 01111011             decimal form
    (short)-123              11111111 10000101             negative form
    (short)32767             01111111 11111111             largest
    (short)-32768            10000000 00000000             smallest
    (short)0173              00000000 01111011             octal form
    (short)0x7B              00000000 01111011             hexadecimal form
```

Type: unsigned short

Description: A non-negative integer.

Size: System dependent. Usually 2 bytes. sizeof(unsigned short) == sizeof(short).

Example Variable Declarations:

```
unsigned short usFirst;
unsigned short usSecond, usThird;
unsigned short int usFourth;
```

Example Literals (assuming size is 2 bytes):

C Literal	Binary Representation	Note
(unsigned short)123	00000000 01111011	decimal form
(unsigned short)65535	11111111 11111111	largest
(unsigned short)0	00000000 00000000	smallest
(unsigned short)0173	00000000 01111011	octal form
(unsigned short)0x7B	00000000 01111011	hexadecimal form

Type: double

Description: A (positive or negative) double-precision floating point number.

Size: System dependent. Often 8 bytes.

Example Variable Declarations:

```
double dFirst;
double dSecond, dThird;
```

Example Literals (assuming size is 8 bytes):

C Literal	Note
123.456	fixed-point notation
1.23456E2	scientific notation
.0123456	fixed-point notation
1.234546E-2	scientific notation with negative exponent
-123.456	fixed-point notation
-1.23456E2	scientific notation with negative mantissa
-.0123456	fixed-point notation
-1.23456E-2	scientific notation with negative mantissa and negative exponent
1.797693E308	largest (approximate)
-1.797693E308	smallest (approximate)
2.225074E-308	closest to 0 (approximate)

Type: float

Description: A (positive or negative) single-precision floating point number.

Size: System dependent. Often 4 bytes. sizeof(float) <= sizeof(double).

Example Variable Declarations:

```
float fFirst;
float fSecond, fThird;
```

Example Literals (assuming size is 4 bytes):

C Literal	Note

```
123.456F              fixed-point notation
1.23456E2F            scientific notation

.0123456F             fixed-point notation
1.234546E-2F          scientific notation with negative exponent

-123.456F             fixed-point notation
-1.23456E2F           scientific notation with negative mantissa

-.0123456F            fixed-point notation
-1.23456E-2F          scientific notation with negative mantissa and negative exponent

3.402823E38F          largest (approximate)
-3.402823E38F         smallest (approximate)
1.175494E-38F         closest to 0 (approximate)
```

Type: long double

Description: A (positive or negative) extended-precision floating point number.

Size: System dependent. Often 12 bytes. sizeof(long double) >= sizeof(double).

Example Variable Declarations:

```
long double ldFirst;
long double ldSecond, ldThird;
```

Example Literals (assuming size is 12 bytes):

```
C Literal             Note

123.456L              fixed-point notation
1.23456E2L            scientific notation

.0123456L             fixed-point notation
1.234546E-2L          scientific notation with negative exponent

-123.456L             fixed-point notation
-1.23456E2L           scientific notation with negative mantissa

-.0123456L            fixed-point notation
-1.23456E-2L          scientific notation with negative mantissa and negative exponent

1.189731E4932L        largest (approximate)
-1.189731E4932L       smallest (approximate)
3.362103E-4932L       closest to 0 (approximate)
```

Differences between C and Java:

Java only:
    boolean, byte

C only:
    unsigned char, unsigned short, unsigned int, unsigned long
    long double

Java: Sizes of all types are **specified**
C: Sizes of all types except char are **system dependent**

Java: char comprises **2** bytes
C: char comprises **1** byte