# Lecture-1
# Introduction

# Agenda

01. Finding the numerical range
02. ABI Memory Model
03. Signed and Unsigned Char
04. scanf and printf in c++
05. Review Questions

# Question 01.

# How to find the numerical range?

| Keyword | Numerical Range Low | High | Digits of Precision | Bytes of Memory |
|---|---|---|---|---|
| `bool` | false | true | n/a | 1 |
| `char` | –128 | 127 | n/a | 1 |
| `short` | –32,768 | 32,767 | n/a | 2 |
| `int` | –2,147,483,648 | 2,147,483,647 | n/a | 4 |
| `long` | –2,147,483,648 | 2,147,483,647 | n/a | 4 |
| `float` | $3.4 \times 10^{-38}$ | $3.4 \times 10^{38}$ | 7 | 4 |
| `double` | $1.7 \times 10^{-308}$ | $1.7 \times 10^{308}$ | 15 | 8 |

| Keyword | Numerical Range Low | High | Bytes of Memory |
|---|---|---|---|
| `unsigned char` | 0 | 255 | 1 |
| `unsigned short` | 0 | 65,535 | 2 |
| `unsigned int` | 0 | 4,294,967,295 | 4 |
| `unsigned long` | 0 | 4,294,967,295 | 4 |

# Finding the numerical range (int)

1. Note the bytes of memory.

2. The range seems to be 2^n. Here n is no of bits.

3. What about zero? The range is from 0 to (2^n)-1.

4. What about negative values? Split the range in half. The range will be from –(2^n)/2 to (2^n)/2-1.

# Finding the numerical range (int)

- #include <limits>
- std::numeric_limits<int>::min(): The minimum value for an int.
- std::numeric_limits<int>::max(): The maximum value for an int.
- std::numeric_limits<double>::max(): The maximum value for a double.

# Question 02.

# Isn't this weird?

| Keyword | Numerical Range Low | High | Digits of Precision | Bytes of Memory |
|---|---|---|---|---|
| `bool` | false | true | n/a | 1 |
| `char` | –128 | 127 | n/a | 1 |
| `short` | –32,768 | 32,767 | n/a | 2 |
| `int` | –2,147,483,648 | 2,147,483,647 | n/a | 4 |
| `long` | –2,147,483,648 | 2,147,483,647 | n/a | 4 |
| `float` | $3.4 \times 10^{-38}$ | $3.4 \times 10^{38}$ | 7 | 4 |
| `double` | $1.7 \times 10^{-308}$ | $1.7 \times 10^{308}$ | 15 | 8 |

| Keyword | Numerical Range Low | High | Bytes of Memory |
|---|---|---|---|
| `unsigned char` | 0 | 255 | 1 |
| `unsigned short` | 0 | 65,535 | 2 |
| `unsigned int` | 0 | 4,294,967,295 | 4 |
| `unsigned long` | 0 | 4,294,967,295 | 4 |

# ABI Memory Models (Application Binary Interface)

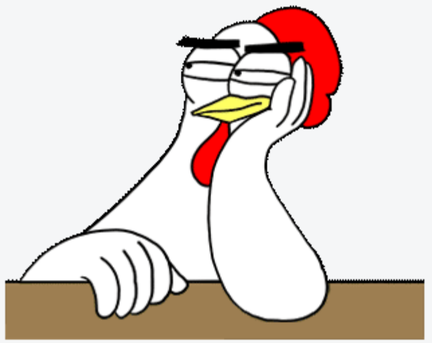
## Overview of Data Model Sizes (in bits)

| Model | int | long | Pointer (P) | Long Long | Usage |
|---|---|---|---|---|---|
| **ILP32** | 32 | 32 | 32 | 64 | 32-bit Linux, Windows, UNIX |
| **LP64** | 32 | 64 | 64 | 64 | 64-bit Linux, macOS, Unix |
| **LLP64** | 32 | 32 | 64 | 64 | 64-bit Windows |
| **ILP64** | 64 | 64 | 64 | 64 | Rare, scientific (e.g., Cray) |

# Question 03.

main.cpp

```cpp
#include <iostream>
#include <cstdio>
using namespace std;
int main() {
    char x = 48;
    char y = -48;
    cout<<"x is: "<<x<<endl;
    cout<<"y is: "<<y<<endl;
}
```



main.cpp

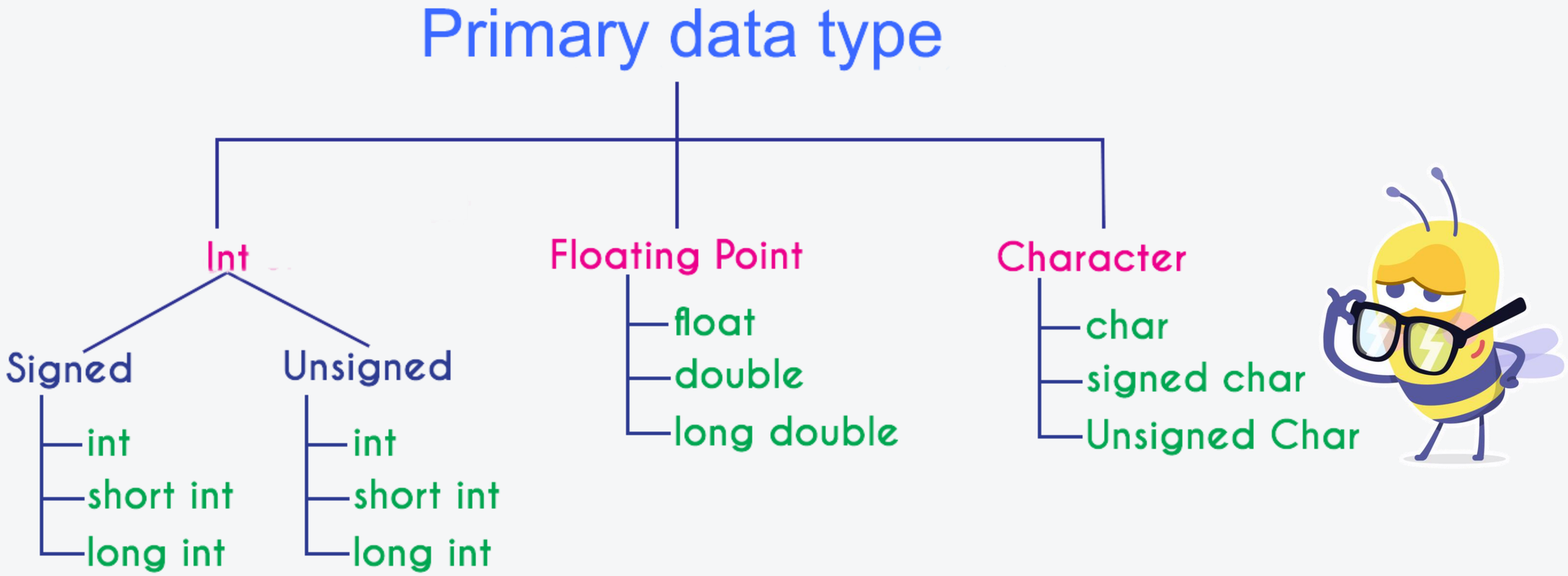
```cpp
#include <iostream>
#include <cstdio>
using namespace std;
int main() {
    char x = 48;
    char y = -48;
    cout<<"x is: "<<x<<endl;
    cout<<"y is: "<<y<<endl;
}
```

```
x is: 0
y is: ╨

Process finished with exit code 0
```

| dec | hex | oct | char | dec | hex | oct | char | dec | hex | oct | char | dec | hex | oct | char |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 000 | NULL | 32 | 20 | 040 | space | 64 | 40 | 100 | @ | 96 | 60 | 140 | ` |
| 1 | 1 | 001 | SOH | 33 | 21 | 041 | ! | 65 | 41 | 101 | A | 97 | 61 | 141 | a |
| 2 | 2 | 002 | STX | 34 | 22 | 042 | " | 66 | 42 | 102 | B | 98 | 62 | 142 | b |
| 3 | 3 | 003 | ETX | 35 | 23 | 043 | # | 67 | 43 | 103 | C | 99 | 63 | 143 | c |
| 4 | 4 | 004 | EOT | 36 | 24 | 044 | $ | 68 | 44 | 104 | D | 100 | 64 | 144 | d |
| 5 | 5 | 005 | ENQ | 37 | 25 | 045 | % | 69 | 45 | 105 | E | 101 | 65 | 145 | e |
| 6 | 6 | 006 | ACK | 38 | 26 | 046 | & | 70 | 46 | 106 | F | 102 | 66 | 146 | f |
| 7 | 7 | 007 | BEL | 39 | 27 | 047 | ' | 71 | 47 | 107 | G | 103 | 67 | 147 | g |
| 8 | 8 | 010 | BS | 40 | 28 | 050 | ( | 72 | 48 | 110 | H | 104 | 68 | 150 | h |
| 9 | 9 | 011 | TAB | 41 | 29 | 051 | ) | 73 | 49 | 111 | I | 105 | 69 | 151 | i |
| 10 | a | 012 | LF | 42 | 2a | 052 | * | 74 | 4a | 112 | J | 106 | 6a | 152 | j |
| 11 | b | 013 | VT | 43 | 2b | 053 | + | 75 | 4b | 113 | K | 107 | 6b | 153 | k |
| 12 | c | 014 | FF | 44 | 2c | 054 | , | 76 | 4c | 114 | L | 108 | 6c | 154 | l |
| 13 | d | 015 | CR | 45 | 2d | 055 | - | 77 | 4d | 115 | M | 109 | 6d | 155 | m |
| 14 | e | 016 | SO | 46 | 2e | 056 | . | 78 | 4e | 116 | N | 110 | 6e | 156 | n |
| 15 | f | 017 | SI | 47 | 2f | 057 | / | 79 | 4f | 117 | O | 111 | 6f | 157 | o |
| 16 | 10 | 020 | DLE | 48 | 30 | 060 | 0 | 80 | 50 | 120 | P | 112 | 70 | 160 | p |
| 17 | 11 | 021 | DC1 | 49 | 31 | 061 | 1 | 81 | 51 | 121 | Q | 113 | 71 | 161 | q |
| 18 | 12 | 022 | DC2 | 50 | 32 | 062 | 2 | 82 | 52 | 122 | R | 114 | 72 | 162 | r |
| 19 | 13 | 023 | DC3 | 51 | 33 | 063 | 3 | 83 | 53 | 123 | S | 115 | 73 | 163 | s |
| 20 | 14 | 024 | DC4 | 52 | 34 | 064 | 4 | 84 | 54 | 124 | T | 116 | 74 | 164 | t |
| 21 | 15 | 025 | NAK | 53 | 35 | 065 | 5 | 85 | 55 | 125 | U | 117 | 75 | 165 | u |
| 22 | 16 | 026 | SYN | 54 | 36 | 066 | 6 | 86 | 56 | 126 | V | 118 | 76 | 166 | v |
| 23 | 17 | 027 | ETB | 55 | 37 | 067 | 7 | 87 | 57 | 127 | W | 119 | 77 | 167 | w |
| 24 | 18 | 030 | CAN | 56 | 38 | 070 | 8 | 88 | 58 | 130 | X | 120 | 78 | 170 | x |
| 25 | 19 | 031 | EM | 57 | 39 | 071 | 9 | 89 | 59 | 131 | Y | 121 | 79 | 171 | y |
| 26 | 1a | 032 | SUB | 58 | 3a | 072 | : | 90 | 5a | 132 | Z | 122 | 7a | 172 | z |
| 27 | 1b | 033 | ESC | 59 | 3b | 073 | ; | 91 | 5b | 133 | [ | 123 | 7b | 173 | { |
| 28 | 1c | 034 | FS | 60 | 3c | 074 | < | 92 | 5c | 134 | \ | 124 | 7c | 174 | \| |
| 29 | 1d | 035 | GS | 61 | 3d | 075 | = | 93 | 5d | 135 | ] | 125 | 7d | 175 | } |
| 30 | 1e | 036 | RS | 62 | 3e | 076 | > | 94 | 5e | 136 | ^ | 126 | 7e | 176 | ~ |
| 31 | 1f | 037 | US | 63 | 3f | 077 | ? | 95 | 5f | 137 | _ | 127 | 7f | 177 | DEL |

# Characters can be signed and unsigned?



# Characters can be signed and unsigned.

```cpp
#include <iostream>
#include <cstdio>
using namespace std;
int main() {

    unsigned char ux = -48;
    unsigned char uy = 48;
    signed char sx = -48;
    signed char sy = 48;

    cout<<"ux is: "<<ux<<endl;
    cout<<"uy is: "<<uy<<endl;
    cout<<"sx is: "<<sx<<endl;
    cout<<"sy is: "<<sy<<endl;

}
```

```
ux is: ╨
uy is: 0
sx is: ╨
sy is: 0

Process finished with exit code 0
```

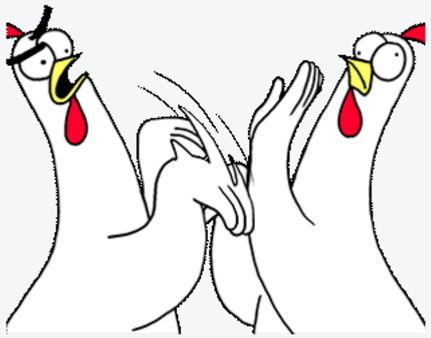
# Question 04.

printf

cout

scanf

cin



```cpp
#include <iostream>
#include <cstdio>

int main() {
    float pi = 3.14159265;

    // Print with 2 digits after decimal
    printf( format: "Pi: %.2f\n", pi);

    // Print with 5 digits after decimal
    printf( format: "Pi: %.5f\n", pi);

}
```

```
Pi: 3.14
Pi: 3.14159

Process finished with exit code 0
```


```
#include <cstdio>

printf("%+8d %8.4E %-4x\n",
        x,y,z);
```

```
#include <iostream>
#include <iomanip>

std::cout << std::setw(8) <<
std::showpos << x <<
std::noshowpos << " " <<
std::setw(8) <<
std::setprecision(4) <<
std::scientific <<
std::uppercase << y <<
std::nouppercase << " " <<
std::setw(4) << std::left <<
std::hex << z << std::endl;
```

# Review Questions

1. Dividing a program into functions
   a. is the key to object-oriented programming.
   b. makes the program easier to conceptualize.
   c. may reduce the size of the program.
   d. makes the program run faster.

1. Dividing a program into functions
   a. is the key to object-oriented programming.
   b. makes the program easier to conceptualize.
   c. may reduce the size of the program.
   d. makes the program run faster.

**B and C**

2. A function name must be followed by _____.
3. A function body is delimited by _____.
4. Why is the `main()` function special?

A C++ instruction that tells the computer to do something is called a _____.

7. An expression

   a. usually evaluates to a numerical value.

   b. indicates the emotional state of the program.

   c. always occurs outside a function.

   d. may be part of a statement.

7. An expression

   a. usually evaluates to a numerical value.

   b. indicates the emotional state of the program.

   c. always occurs outside a function.

   d. may be part of a statement.

**A and D**

8. Specify how many bytes are occupied by the following data types in a 32-bit system:
   a. Type `int`
   b. Type `long double`
   c. Type `float`
   d. Type `long`

9. True or false: A variable of type `char` can hold the value 301.

10. What kind of program elements are the following?

    a. 12

    b. `'a'`

    c. 4.28915

    d. `JungleJim`

    e. `JungleJim()`

12. True or false: In an assignment statement, the value on the left of the equal sign is always equal to the value on the right.

14. What header file must you `#include` with your source file to use `cout` and `cin`?

16. What header file must you `#include` with your program to use `setw`?

17. Two exceptions to the rule that the compiler ignores whitespace are _____ and _____.

23. Assuming `var1` starts with the value 20, what will the following code fragment print out?

```
cout << var1--;
cout << ++var1;
```

18. True or false: It's perfectly all right to use variables of different data types in the same arithmetic expression.

# Thank You!