C++

Prepared by: Mohamed Ayman

facebook.com/sw.eng.MohamedAyman

sw.eng.MohamedAyman@gmail.com

wuzzuf.net/me/engMohamedAyman

codeforces.com/profile/Mohamed_Ayman

Variable Types

Outline

1) Standard Data Types
2) Storage of Data Types
3) Variable Types
4) L-value vs. R-value
5) Data Type Conversion

Standard Data Types

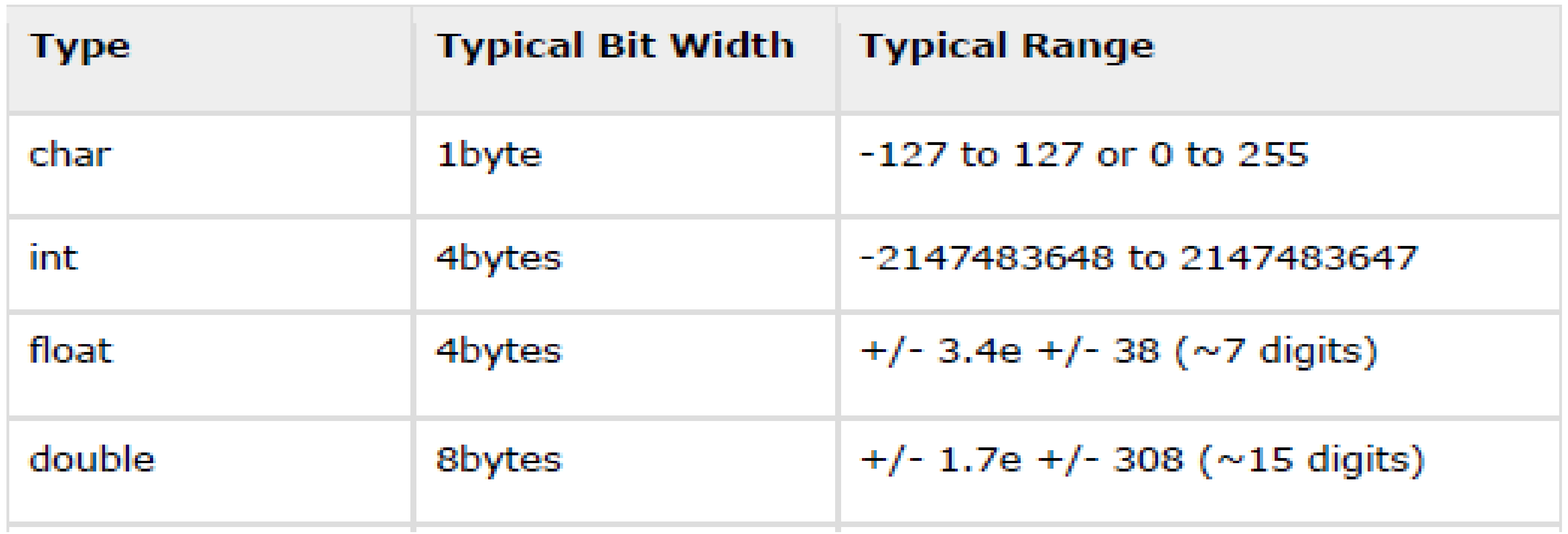
- While writing program in any language, you need to use various variables to store various information. Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

- Based on the data type of a variable, the operating system allocates memory and decides what can be stored in the reserved memory. Primitive Built-in Types C++ offers the programmer a rich assortment of built-in as well as user defined data types.

Standard Data Types

- Some Primitive data types in c++: int, float, double, char, bool

The following table shows the variable type, how much memory it takes to store the value in memory, and what is maximum and minimum value which can be stored in such type of variables.

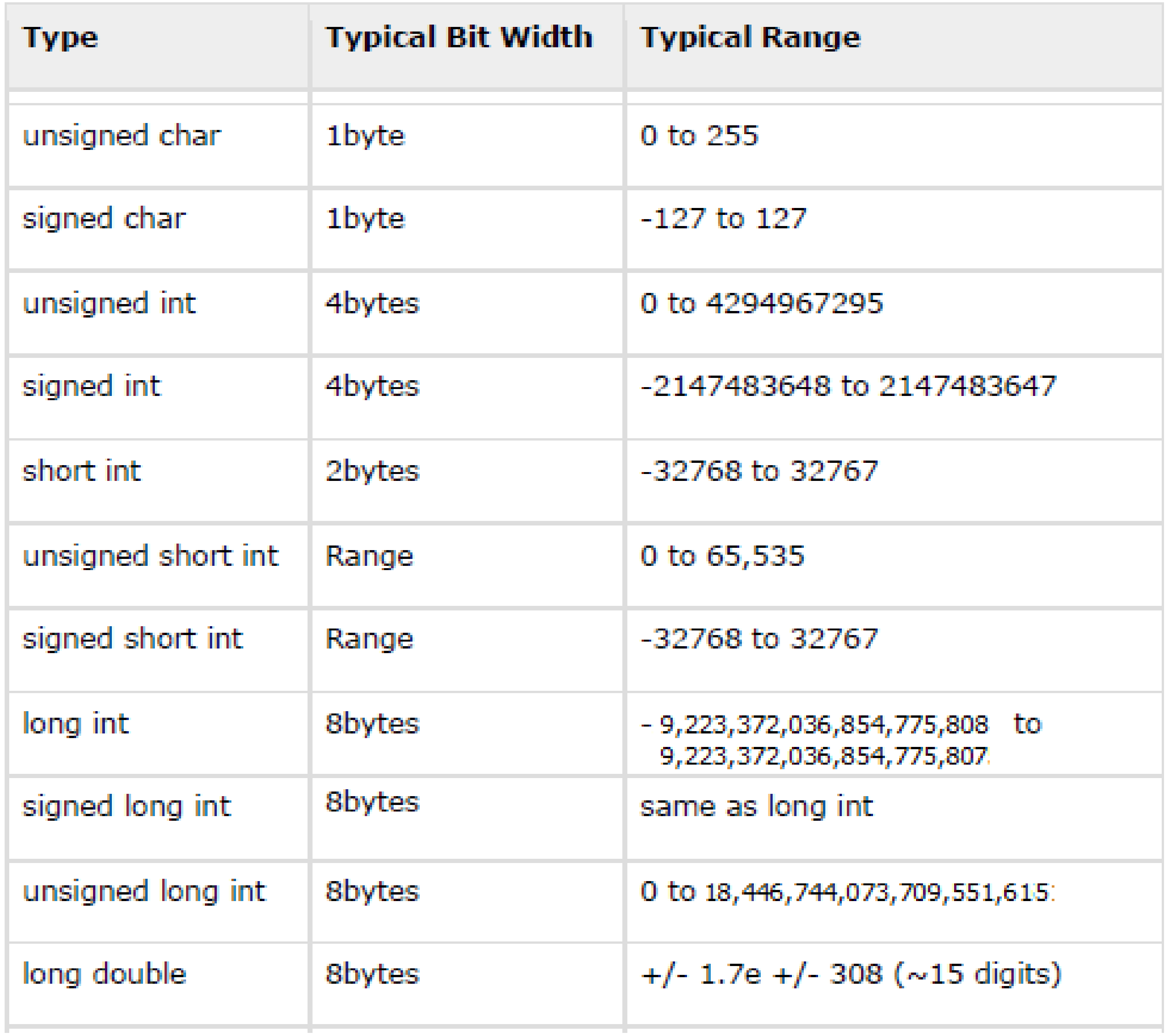


Type	Typical Bit Width	Typical Range
char	1byte	-127 to 127 or 0 to 255
int	4bytes	-2147483648 to 2147483647
float	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	8bytes	+/- 1.7e +/- 308 (~15 digits)

Standard Data Types

- Several of the basic types can be modified using one or more of these type modifiers:

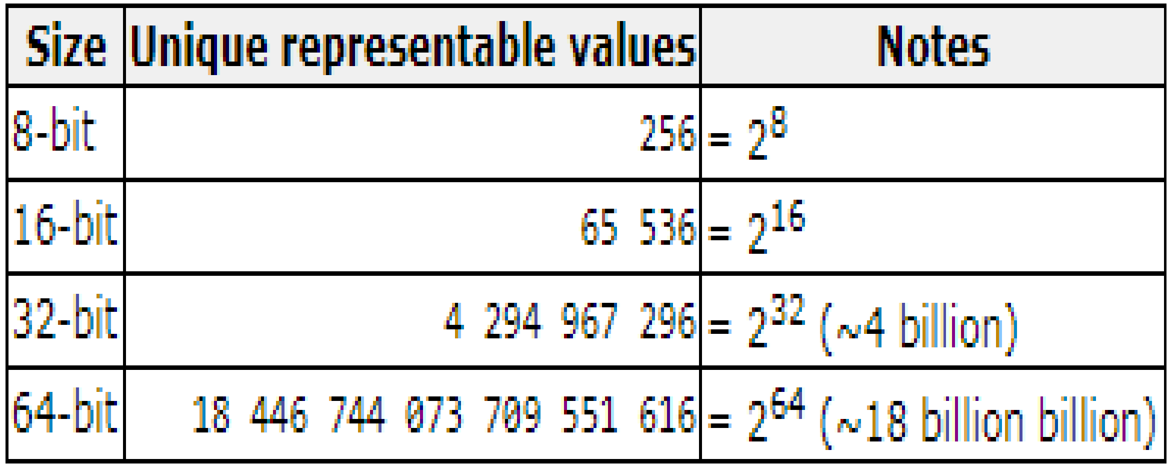
  signed, unsigned, short, long



Type	Typical Bit Width	Typical Range
unsigned char	1byte	0 to 255
signed char	1byte	-127 to 127
unsigned int	4bytes	0 to 4294967295
signed int	4bytes	-2147483648 to 2147483647
short int	2bytes	-32768 to 32767
unsigned short int	Range	0 to 65,535
signed short int	Range	-32768 to 32767
long int	8bytes	- 9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
signed long int	8bytes	same as long int
unsigned long int	8bytes	0 to 18,446,744,073,709,551,615
long double	8bytes	+/- 1.7e +/- 308 (~15 digits)

Storage of Data Types

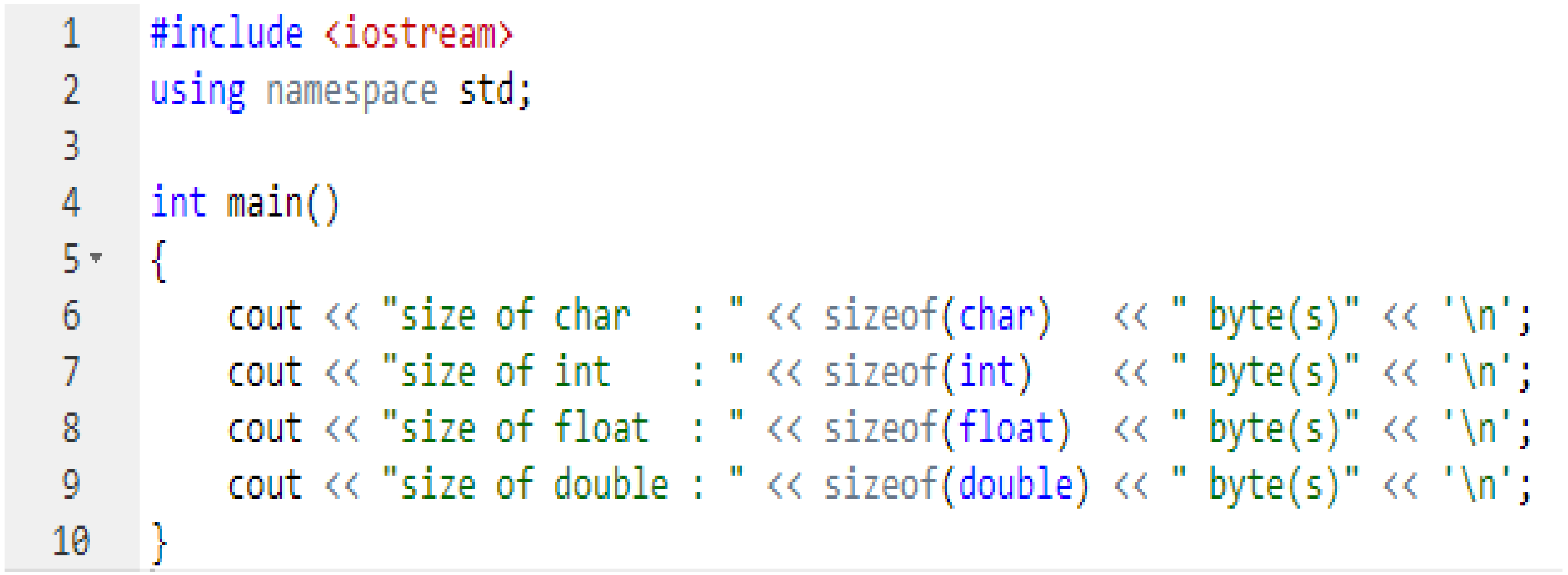
- Bit:
  - A "bit", like an atom, the smallest unit of storage
  - A bit stores just a 0 or 1
  - In the computer it's all 0's and 1's ... bits
  - Anything with two separate states can store 1 bit
  - A bit is too small to be much use
  - Group 8 bits together to make 1 byte
- Byte:
  - One byte = grouping of 8 bits
- Kilobyte, KB, about 1 thousand bytes
- Megabyte, MB, about 1 million bytes
- Gigabyte, GB, about 1 billion bytes
- Terabyte, TB, about 1 trillion bytes



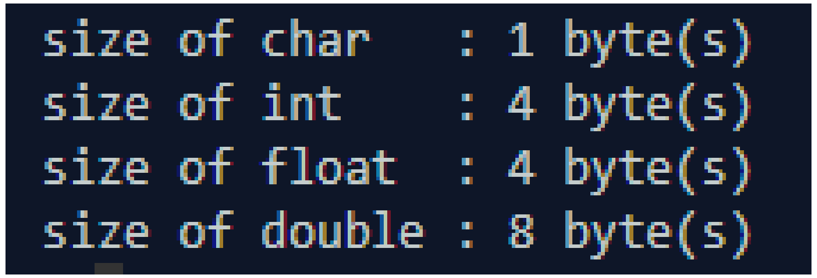
Size	Unique representable values	Notes
8-bit	256	= 2^8
16-bit	65 536	= 2^{16}
32-bit	4 294 967 296	= 2^{32} (~4 billion)
64-bit	18 446 744 073 709 551 616	= 2^{64} (~18 billion billion)

Storage of Data Types Example

- Source code: https://repl.it/repls/ScratchyPunctualThrasher



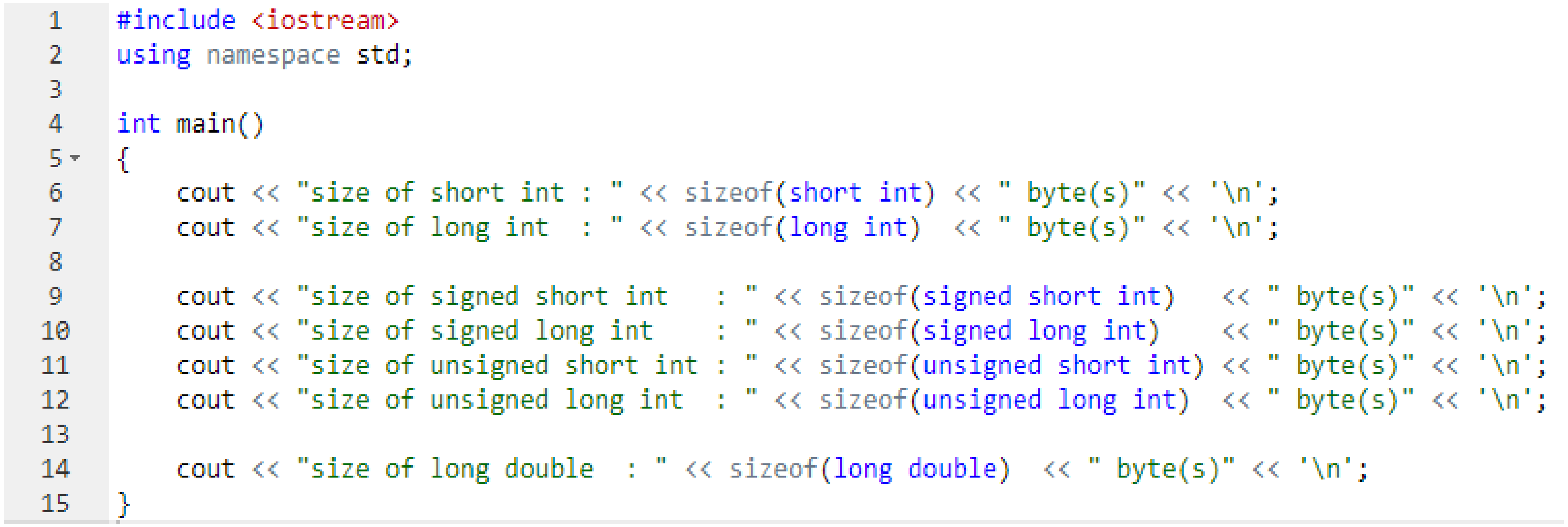
```cpp
#include <iostream>
using namespace std;

int main()
{
    cout << "size of char   : " << sizeof(char)   << " byte(s)" << '\n';
    cout << "size of int    : " << sizeof(int)    << " byte(s)" << '\n';
    cout << "size of float  : " << sizeof(float)  << " byte(s)" << '\n';
    cout << "size of double : " << sizeof(double) << " byte(s)" << '\n';
}
```



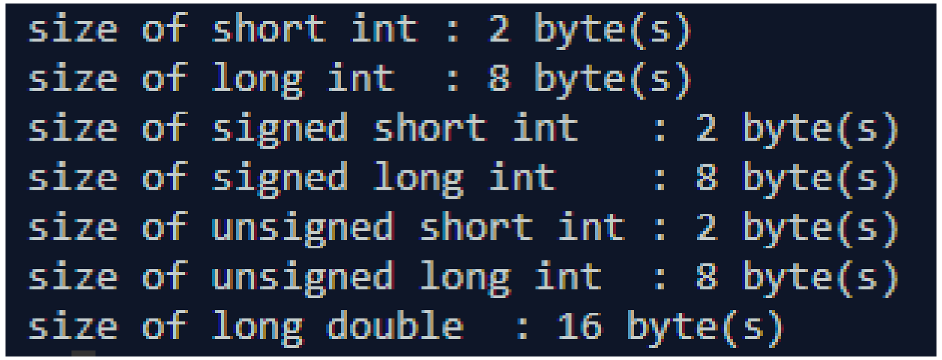
```
size of char   : 1 byte(s)
size of int    : 4 byte(s)
size of float  : 4 byte(s)
size of double : 8 byte(s)
```

Storage of Data Types Example

- Source code: https://repl.it/repls/LovableFearfulTrumpeterswan



```cpp
#include <iostream>
using namespace std;

int main()
{
    cout << "size of short int : " << sizeof(short int) << " byte(s)" << '\n';
    cout << "size of long int  : " << sizeof(long int)  << " byte(s)" << '\n';

    cout << "size of signed short int   : " << sizeof(signed short int)   << " byte(s)" << '\n';
    cout << "size of signed long int    : " << sizeof(signed long int)    << " byte(s)" << '\n';
    cout << "size of unsigned short int : " << sizeof(unsigned short int) << " byte(s)" << '\n';
    cout << "size of unsigned long int  : " << sizeof(unsigned long int)  << " byte(s)" << '\n';

    cout << "size of long double  : " << sizeof(long double)  << " byte(s)" << '\n';
}
```



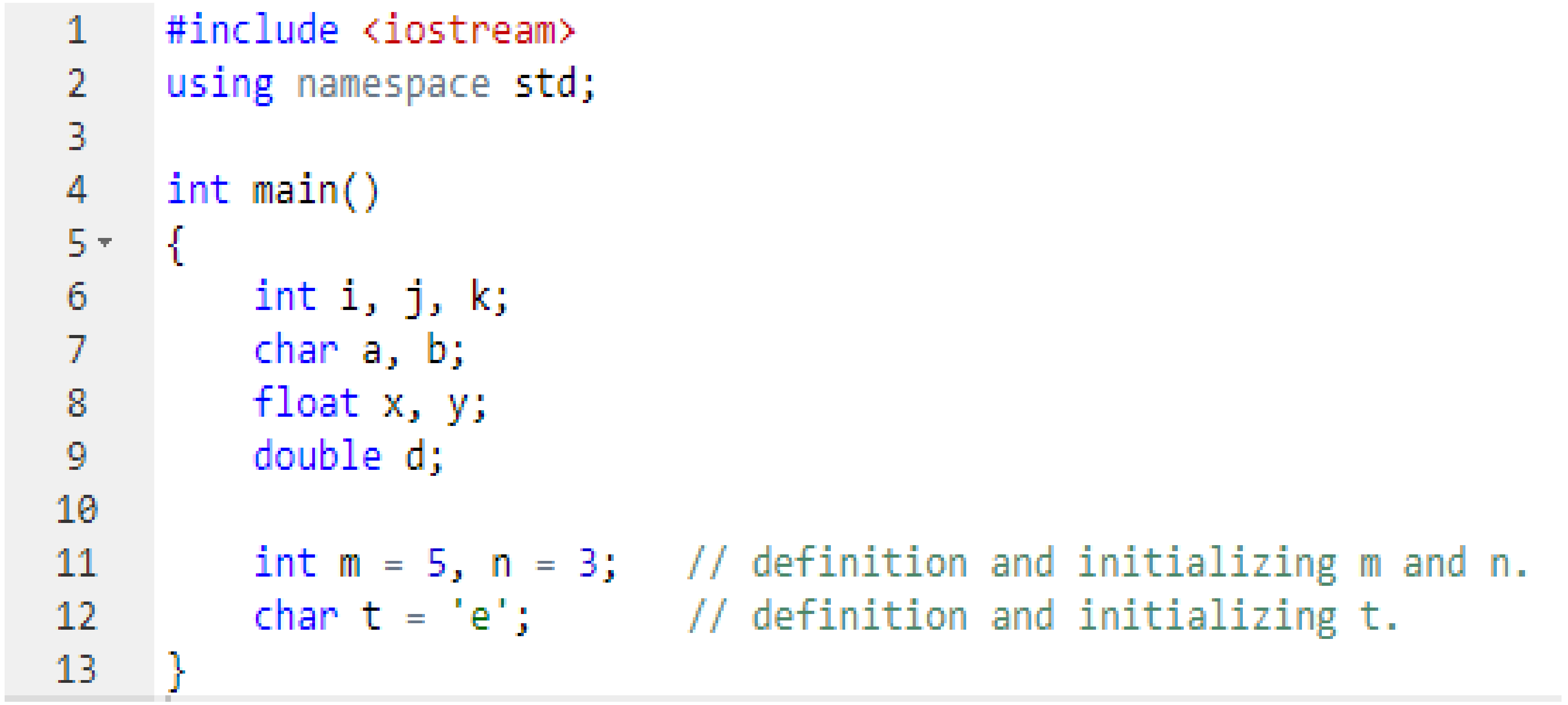
```
size of short int : 2 byte(s)
size of long int  : 8 byte(s)
size of signed short int   : 2 byte(s)
size of signed long int    : 8 byte(s)
size of unsigned short int : 2 byte(s)
size of unsigned long int  : 8 byte(s)
size of long double  : 16 byte(s)
```

Variable Types

- A variable definition tells the compiler where and how much storage to create for the variable. A variable definition specifies a data type, and contains a list of one or more variables of that type as follows:

Variable Types Example

- Source code: https://repl.it/repls/FarawayAuthenticMarbledmurrelet



```cpp
#include <iostream>
using namespace std;

int main()
{
    int i, j, k;
    char a, b;
    float x, y;
    double d;

    int m = 5, n = 3;   // definition and initializing m and n.
    char t = 'e';       // definition and initializing t.
}
```

L-values and R-values

There are two kinds of expressions in C++:

- L-value : Expressions that refer to a memory location is called "L-value" expression. An L-value may appear as either the left-hand or right-hand side of an assignment.

- R-value : The term R-value refers to a data value that is stored at some address in memory. An R-value is an expression that cannot have a value assigned to it which means an R-value

L-values and R-values Example

Variables are lvalues and so may appear on the left-hand side of an assignment. Numeric literals are rvalues and so may not be assigned and cannot appear on the left-hand side. Following is a valid statement:

```
int g = 20;
```

But the following is not a valid statement and would generate compile-time error:

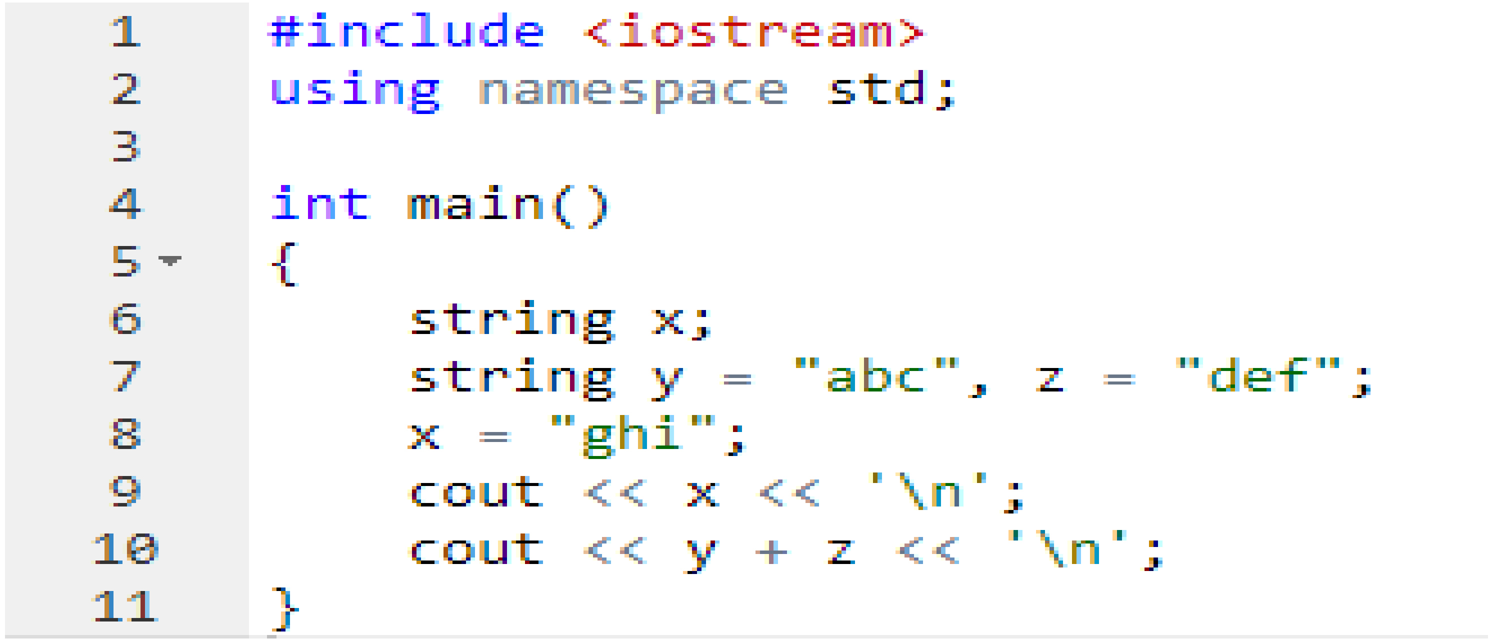
```
10 = 20;
```

String Definition

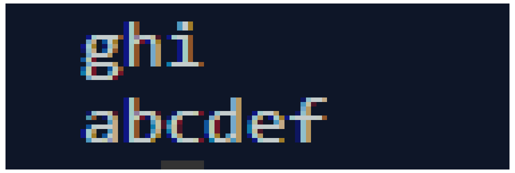
- This string is actually a one-dimensional array of characters which is terminated by a null character '\0'. Thus a null-terminated string contains the characters that comprise the string followed by a null.

String Example

- Source code: https://repl.it/repls/FewJealousBluebird



```cpp
#include <iostream>
using namespace std;

int main()
{
	string x;
	string y = "abc", z = "def";
	x = "ghi";
	cout << x << '\n';
	cout << y + z << '\n';
}
```



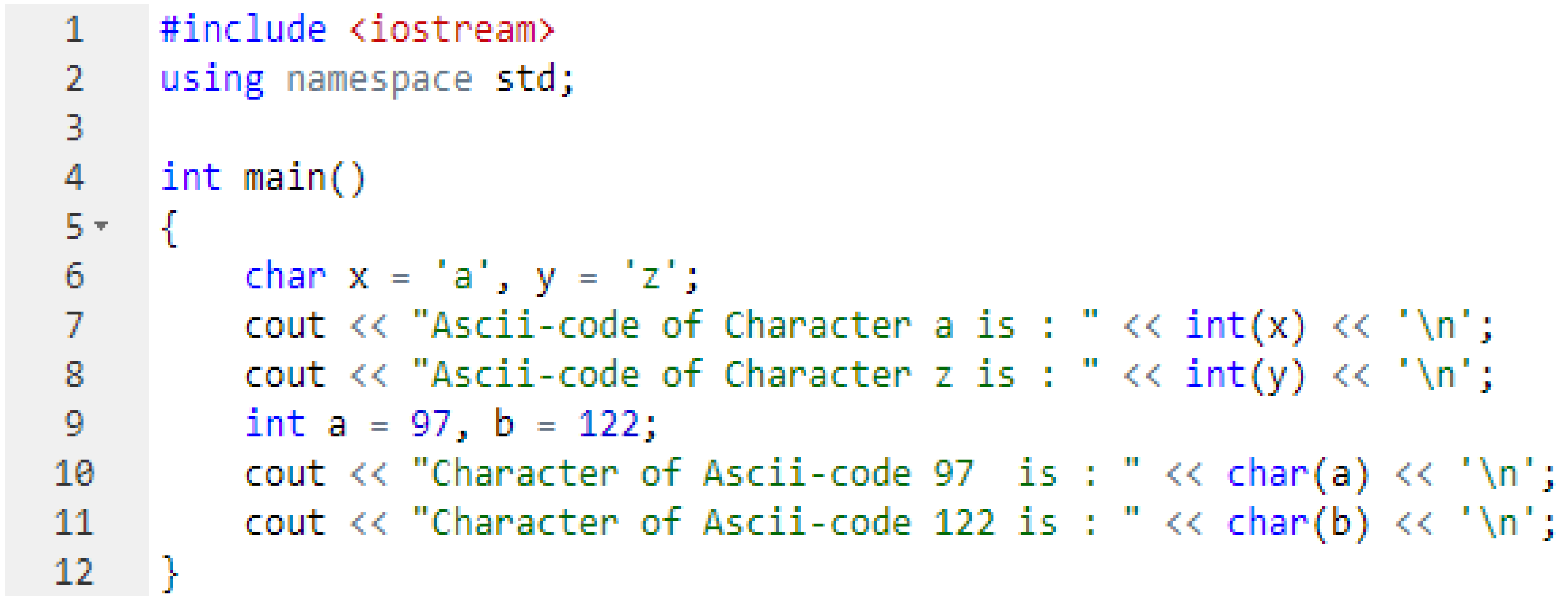
```
ghi
abcdef
```

Data Type Conversion

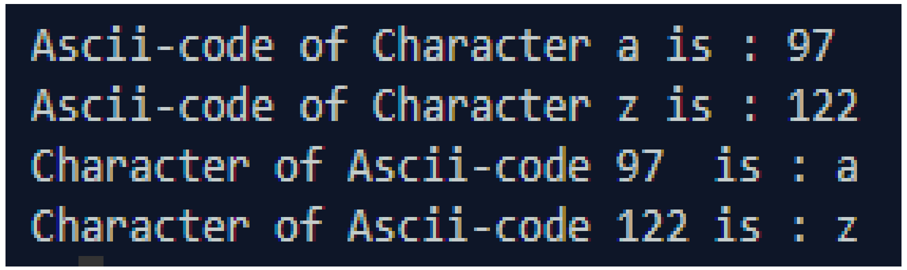
- Sometimes, you may need to perform conversions between the built-in types, you simply use the type-name as a function.
- There are several built-in functions to perform conversion from one data type to another.
- These functions return a new object representing the converted value.

Data Type Conversion Example

- Source code: https://repl.it/repls/FamiliarNewJenny



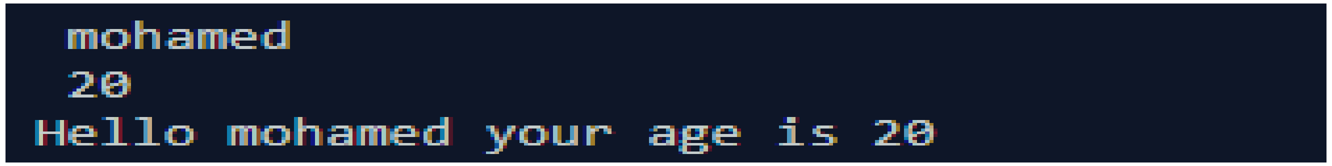
```cpp
#include <iostream>
using namespace std;

int main()
{
    char x = 'a', y = 'z';
    cout << "Ascii-code of Character a is : " << int(x) << '\n';
    cout << "Ascii-code of Character z is : " << int(y) << '\n';
    int a = 97, b = 122;
    cout << "Character of Ascii-code 97  is : " << char(a) << '\n';
    cout << "Character of Ascii-code 122 is : " << char(b) << '\n';
}
```



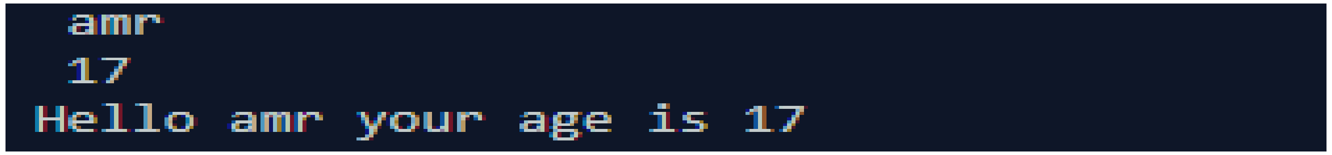
```
Ascii-code of Character a is : 97
Ascii-code of Character z is : 122
Character of Ascii-code 97  is : a
Character of Ascii-code 122 is : z
```

Practice

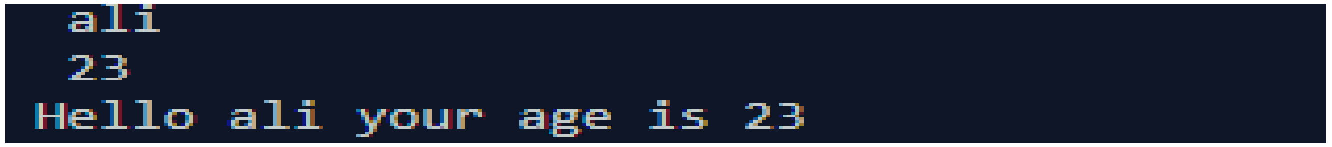
- Take as an input name and age of user in separate lines then print :
  hello "userName" your age is "userAge"
- Test Cases:



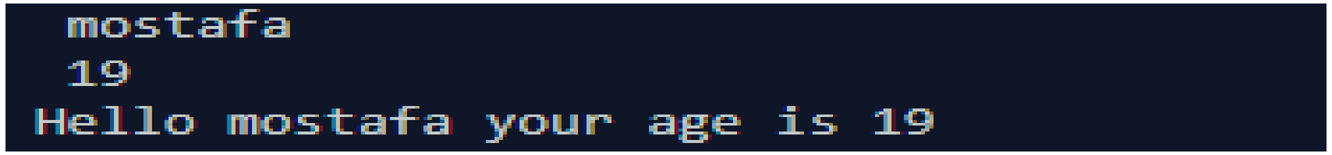
```
 mohamed
 20
Hello mohamed your age is 20
```



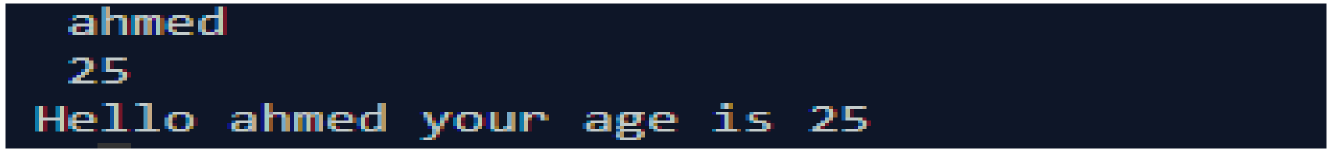
```
 amr
 17
Hello amr your age is 17
```



```
 ali
 23
Hello ali your age is 23
```



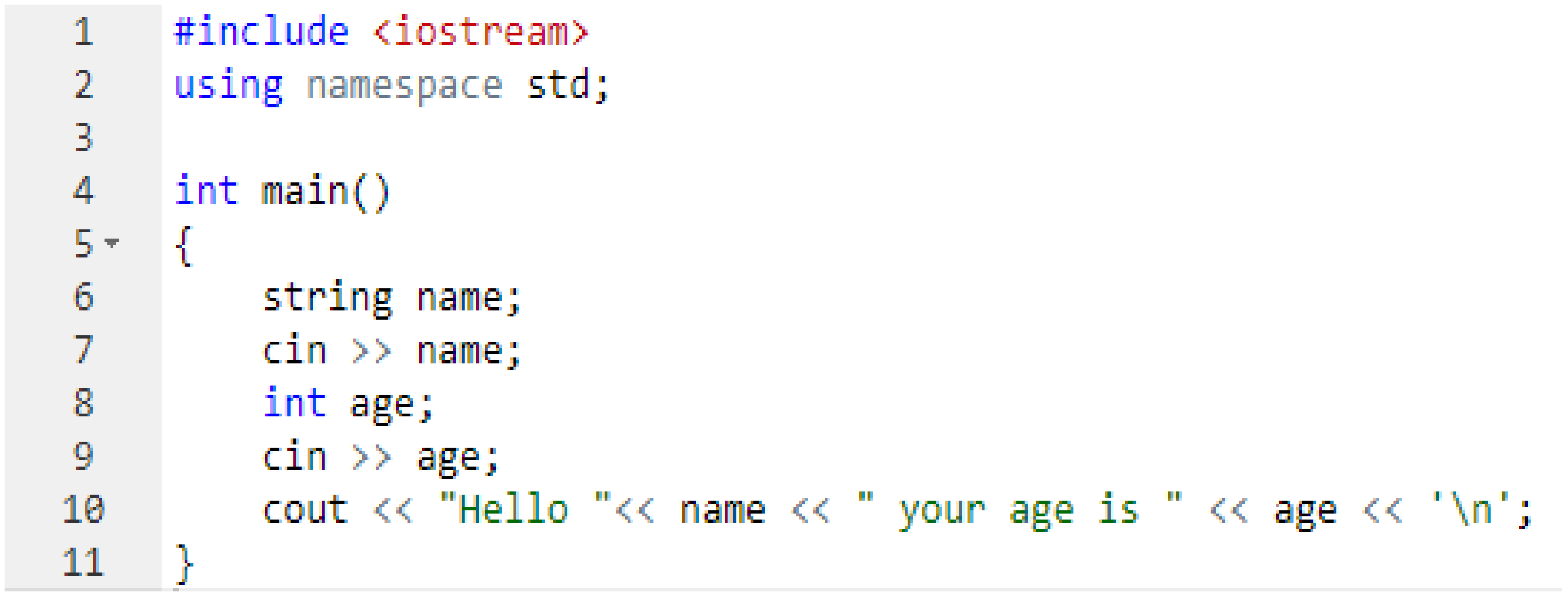
```
 mostafa
 19
Hello mostafa your age is 19
```



```
 ahmed
 25
Hello ahmed your age is 25
```

Practice Solution

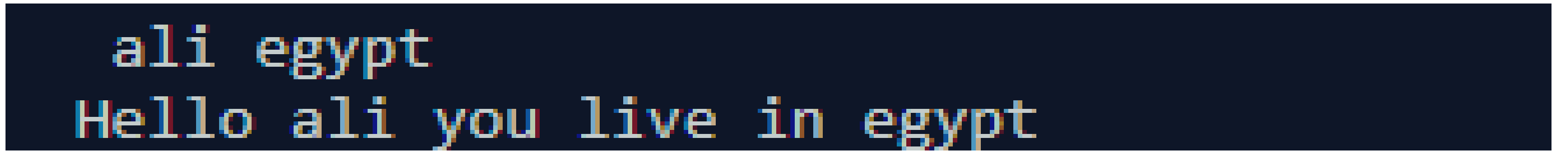
- Source Code: https://repl.it/repls/SuperbStaleCockroach



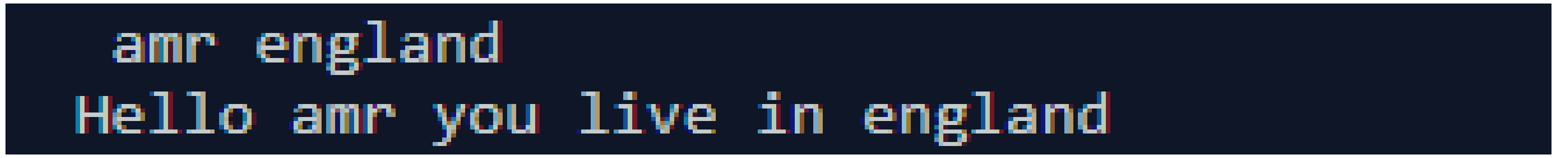
```cpp
#include <iostream>
using namespace std;

int main()
{
    string name;
    cin >> name;
    int age;
    cin >> age;
    cout << "Hello "<< name << " your age is " << age << '\n';
}
```

Practice

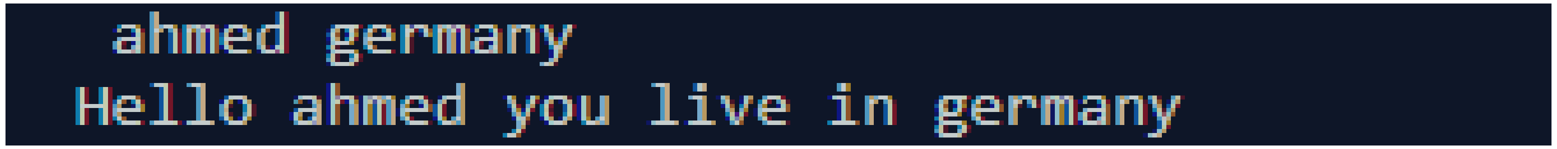
- Take as an input name and city of user in same line then print
  hello “userName” you live in “userCity”
- Test Cases:



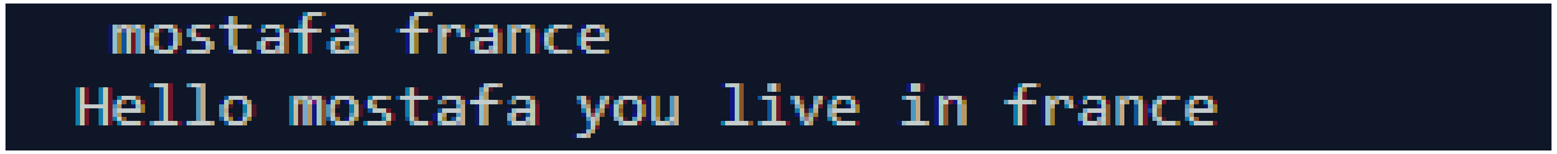
```
 ali egypt
Hello ali you live in egypt
```



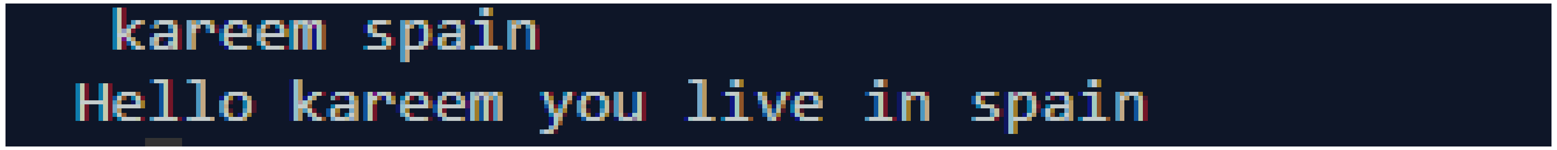
```
 amr england
Hello amr you live in england
```



```
 ahmed germany
Hello ahmed you live in germany
```



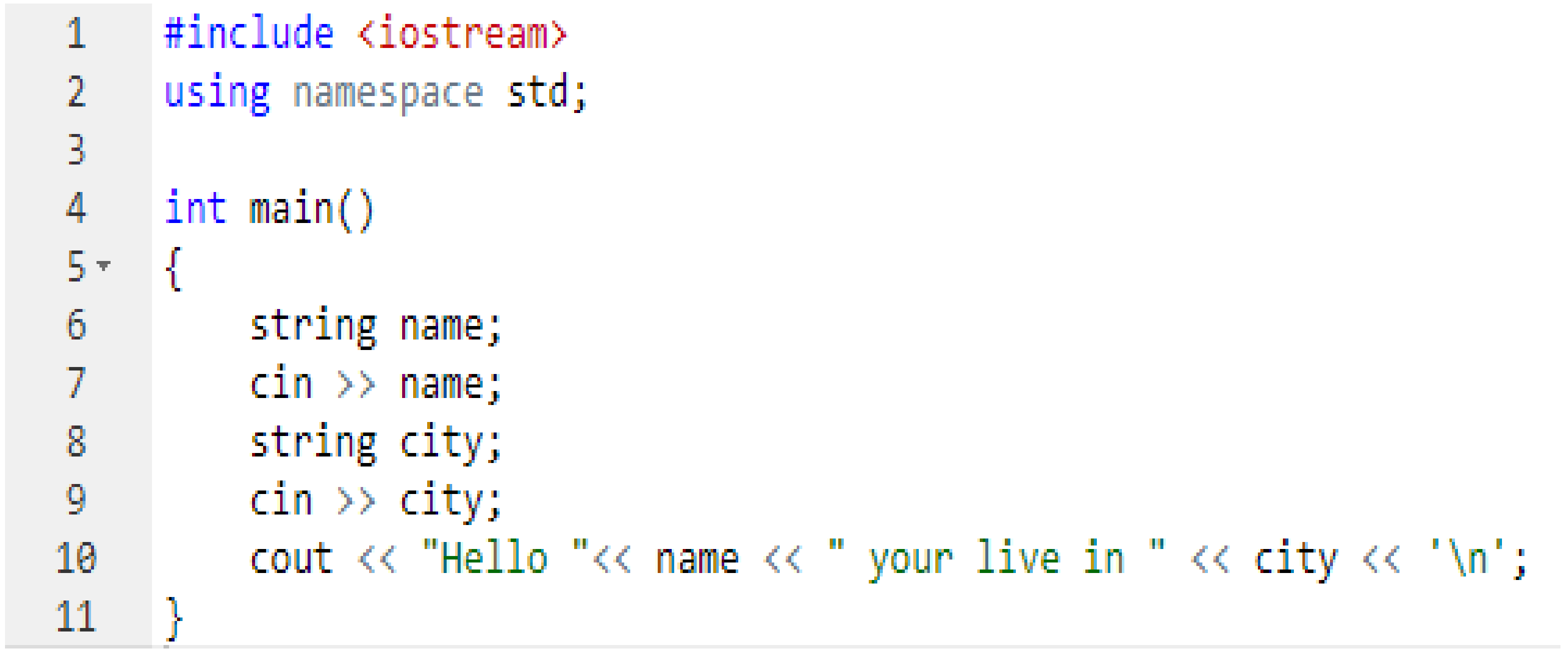
```
 mostafa france
Hello mostafa you live in france
```



```
 kareem spain
Hello kareem you live in spain
```

Practice Solution

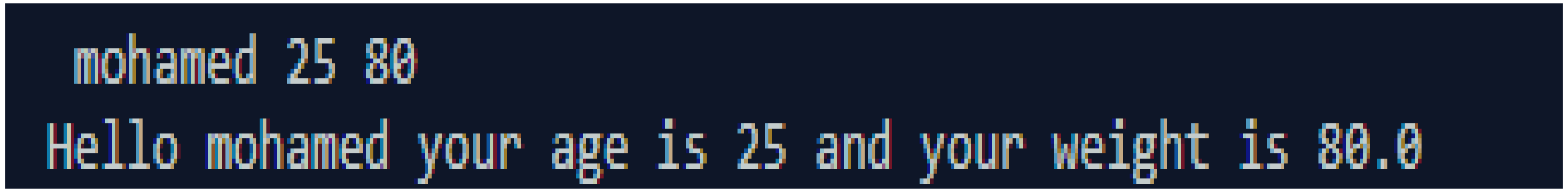
- Source Code: https://repl.it/repls/AwareKosherHatchetfish



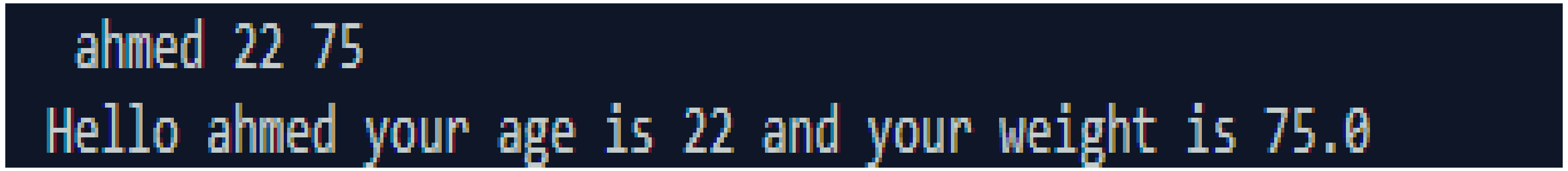
```cpp
#include <iostream>
using namespace std;

int main()
{
    string name;
    cin >> name;
    string city;
    cin >> city;
    cout << "Hello "<< name << " your live in " << city << '\n';
}
```

Practice

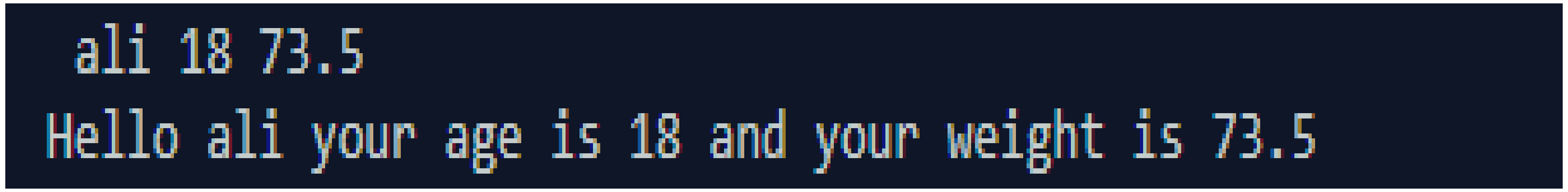
- Take as an input name and age and weight of user in same line then print : hello "userName" your age is "userAge" and your weight is "userWeight"
- Test Cases:



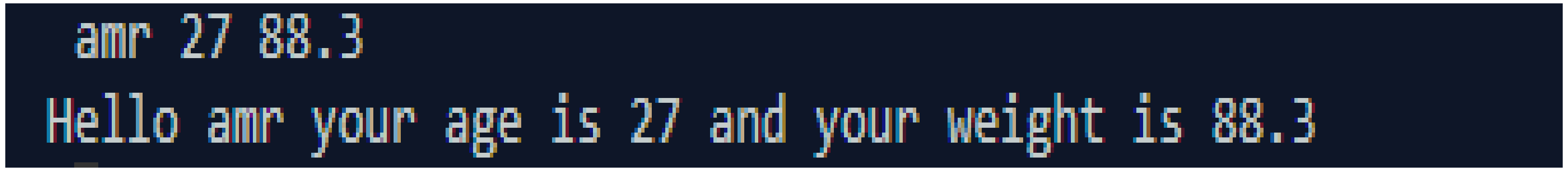
```
 mohamed 25 80
Hello mohamed your age is 25 and your weight is 80.0
```



```
 ahmed 22 75
Hello ahmed your age is 22 and your weight is 75.0
```



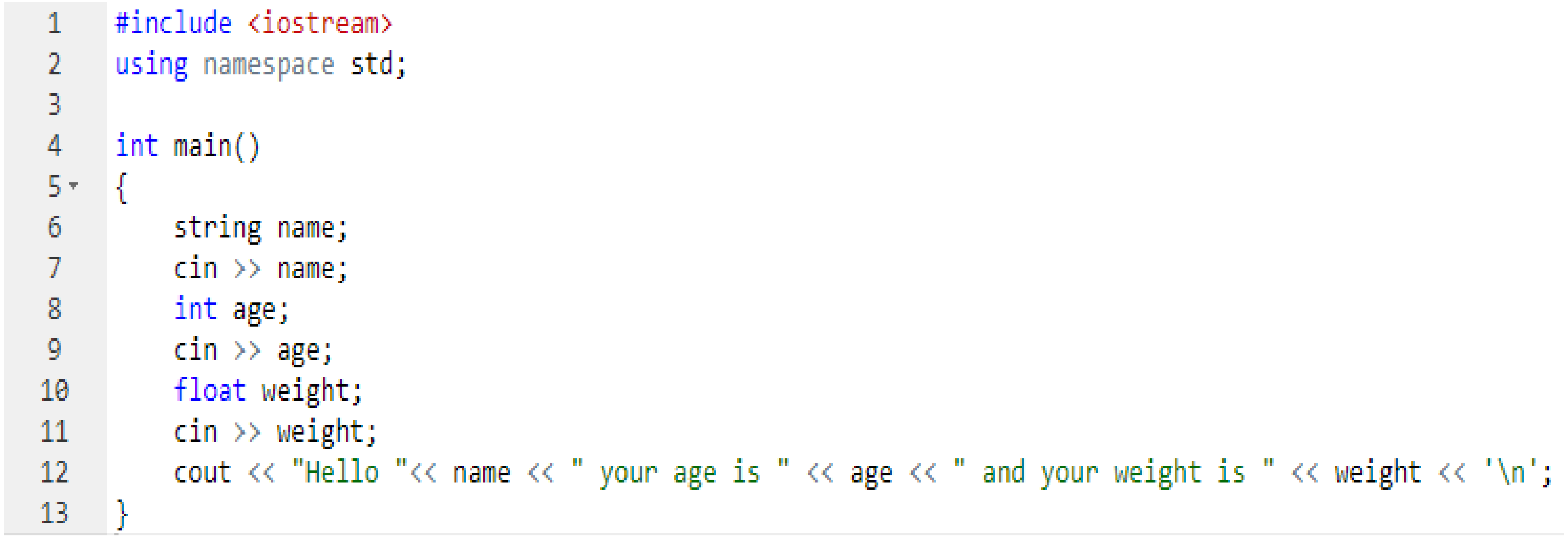
```
 ali 18 73.5
Hello ali your age is 18 and your weight is 73.5
```



```
 amr 27 88.3
Hello amr your age is 27 and your weight is 88.3
```

Practice Solution

- Source Code: https://repl.it/repls/KeenGrandioseWolf



```cpp
#include <iostream>
using namespace std;

int main()
{
    string name;
    cin >> name;
    int age;
    cin >> age;
    float weight;
    cin >> weight;
    cout << "Hello "<< name << " your age is " << age << " and your weight is " << weight << '\n';
}
```

Questions ?

References

Online Courses YouTube playlists:	http://bit.ly/2kAPL5K
C++ Documentation	http://bit.ly/1flmcHO
CPP For School	http://bit.ly/2kifMdj
C++ Language Tutorial	http://bit.ly/1kyBMdz
C++ Language Tutorial	http://bit.ly/2rzE4hQ
C++ Tutorial Point	http://bit.ly/2BGFeO0
Fundamentals of C++ Programming	http://bit.ly/2rJHhyI
Teach Yourself C++ in 21 Days	http://bit.ly/1JXhDtL
A Complete Guide to Programming in C++	http://bit.ly/2dVkGY9