C++

پریسا حامد روح بخش

موسسه ی پارس پژوهان

فصل دوم

- **Variables**
- **data types**
- **Identifiers**
- **Constants**

Variables

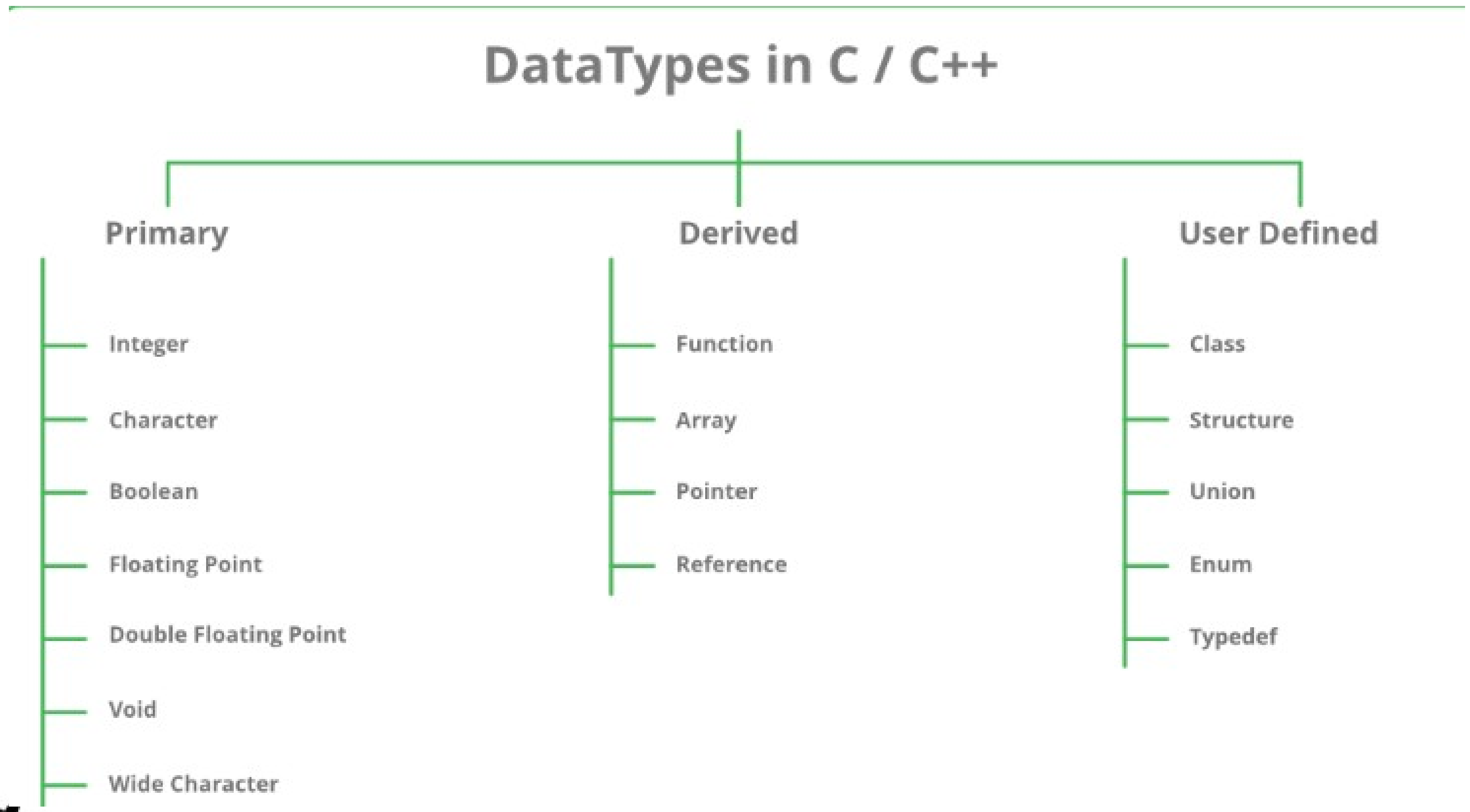
- Creating a variable reserves a memory location, or a space in memory for storing values. The compiler requires that you provide a data type for each variable you declare.

data types

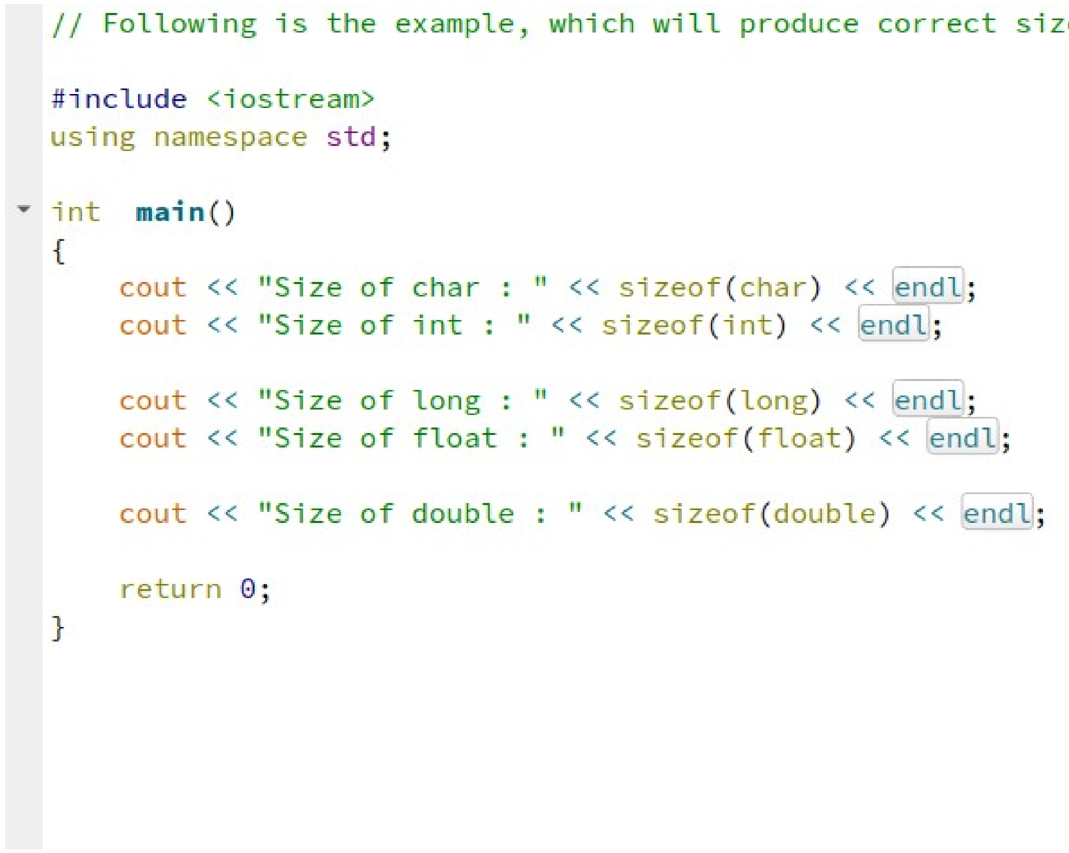
Data Type	Size	Description
`boolean`	1 byte	Stores true or false values
`char`	1 byte	Stores a single character/letter/number, or ASCII values
`int`	2 or 4 bytes	Stores whole numbers, without decimals
`float`	4 bytes	Stores fractional numbers, containing one or more decimals. Sufficient for storing 7 decimal digits
`double`	8 bytes	Stores fractional numbers, containing one or more decimals. Sufficient for storing 15 decimal digits

```
int myNum = 9;
double myDoubleNum = 8.99;
char myLetter = 'A';
bool myBool = false;
string myText = "Hello World";
```

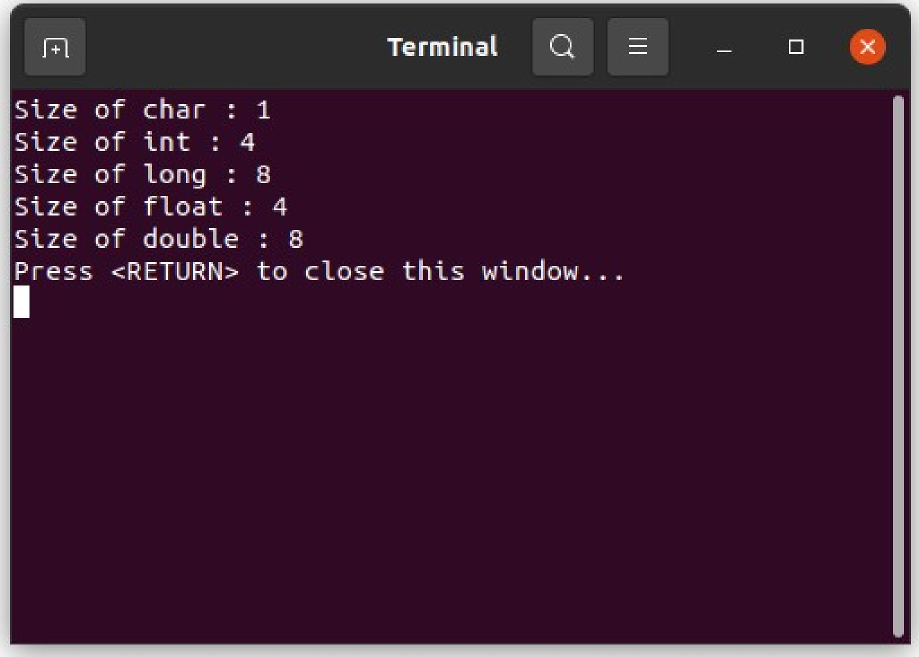
data types


DataTypes in C / C++
Primary
Integer
Character
Boolean
Floating Point
Double Floating Point
Void
Wide Character
Derived
Function
Array
Pointer
Reference
User Defined
Class
Structure
Union
Enum
Typedef

sizeof()



```cpp
// Following is the example, which will produce correct size of various data types on your computer.

#include <iostream>
using namespace std;

int  main()
{
    cout << "Size of char : " << sizeof(char) << endl;
    cout << "Size of int : " << sizeof(int) << endl;

    cout << "Size of long : " << sizeof(long) << endl;
    cout << "Size of float : " << sizeof(float) << endl;

    cout << "Size of double : " << sizeof(double) << endl;

    return 0;
}
```



```
Size of char : 1
Size of int : 4
Size of long : 8
Size of float : 4
Size of double : 8
Press <RETURN> to close this window...
```

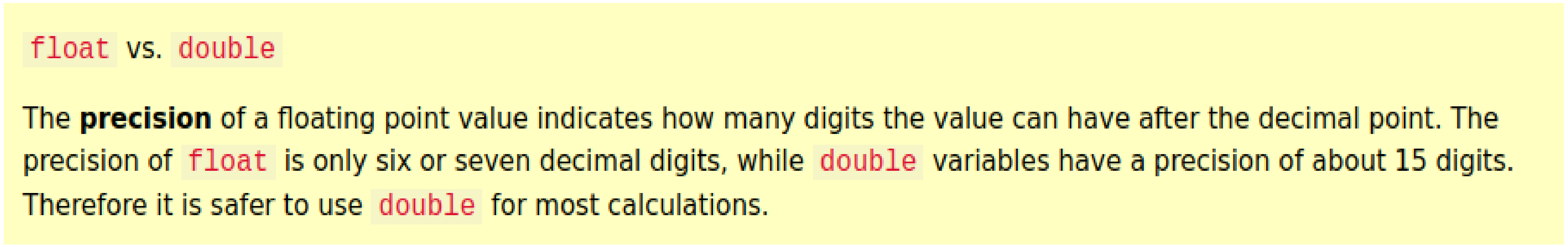
Variables

- Define all variables with a name and a data type before using them in a program. In cases in which you have multiple variables of the same type, it's possible to define them in one declaration, separating them with commas

```
int a, b;
// defines two variables of type int
```

Numeric Types

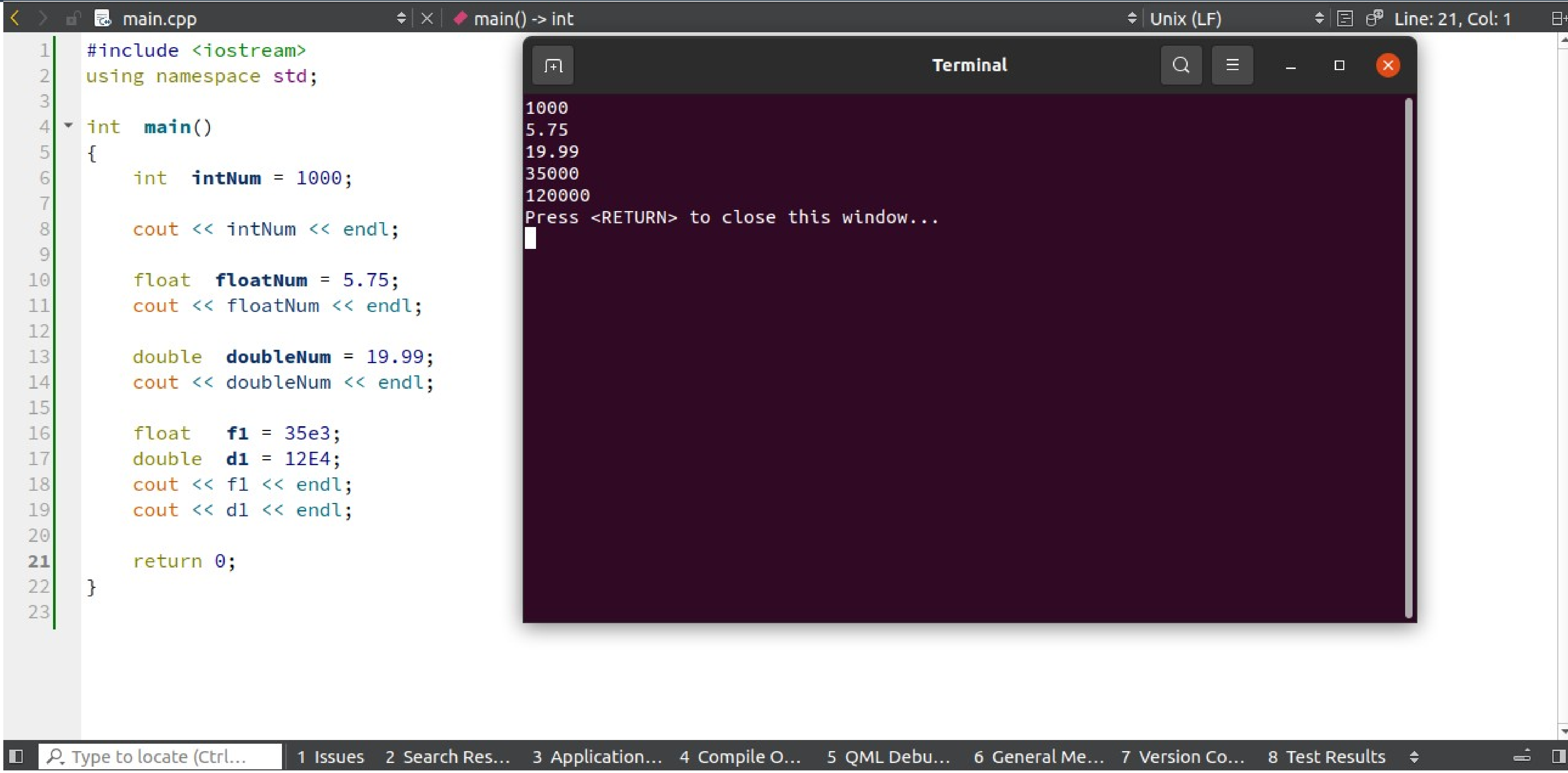
use int (Integer) when you need to store a whole number without decimals, like 35 or 1000, and float or double when you need a floating point number (with decimals), like 9.99 or 3.14515.



`float` vs. `double`

The **precision** of a floating point value indicates how many digits the value can have after the decimal point. The precision of `float` is only six or seven decimal digits, while `double` variables have a precision of about 15 digits. Therefore it is safer to use `double` for most calculations.

Example



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

int  main()
{
    int  intNum = 1000;

    cout << intNum << endl;

    float  floatNum = 5.75;
    cout << floatNum << endl;

    double  doubleNum = 19.99;
    cout << doubleNum << endl;

    float   f1 = 35e3;
    double  d1 = 12E4;
    cout << f1 << endl;
    cout << d1 << endl;

    return 0;
}
```

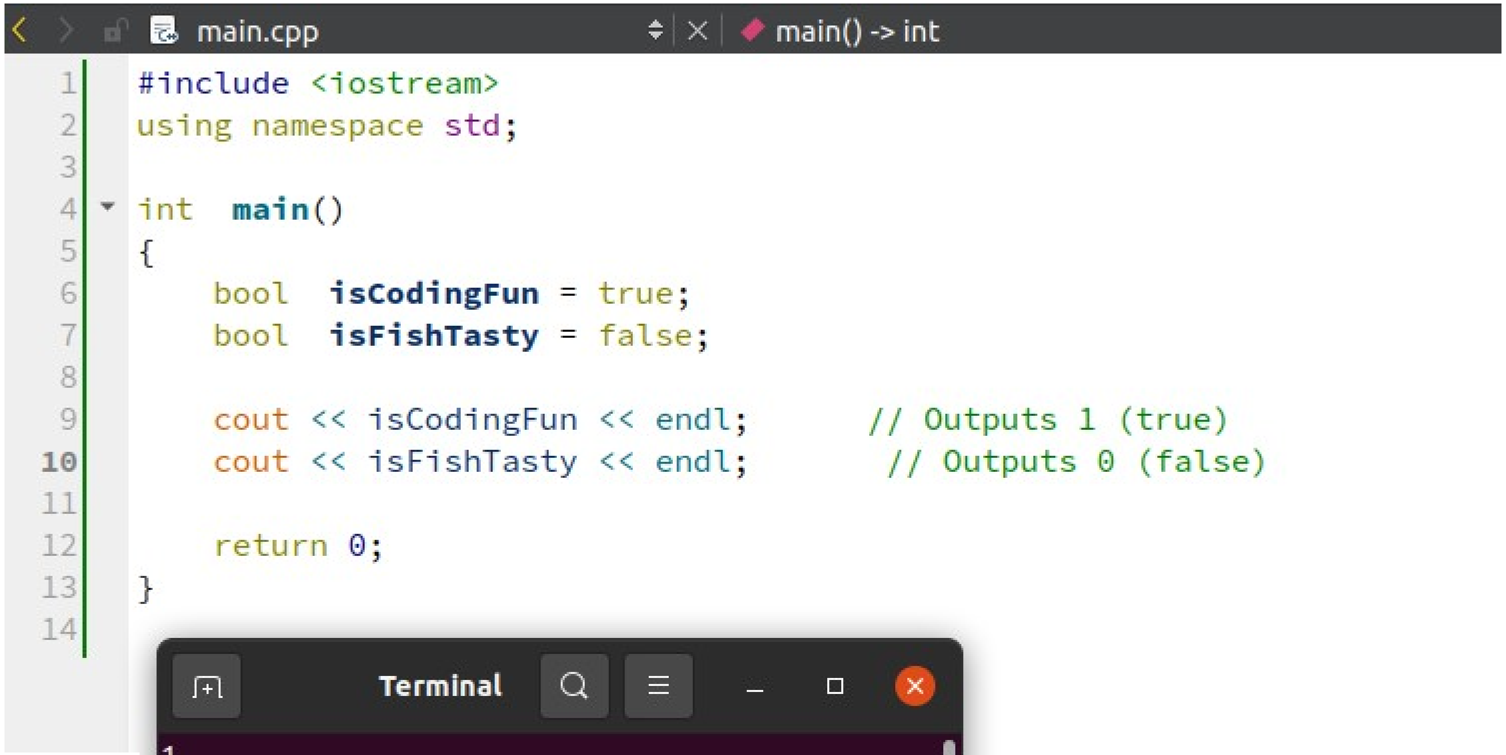
```
1000
5.75
19.99
35000
120000
Press <RETURN> to close this window...
```

Boolean Types

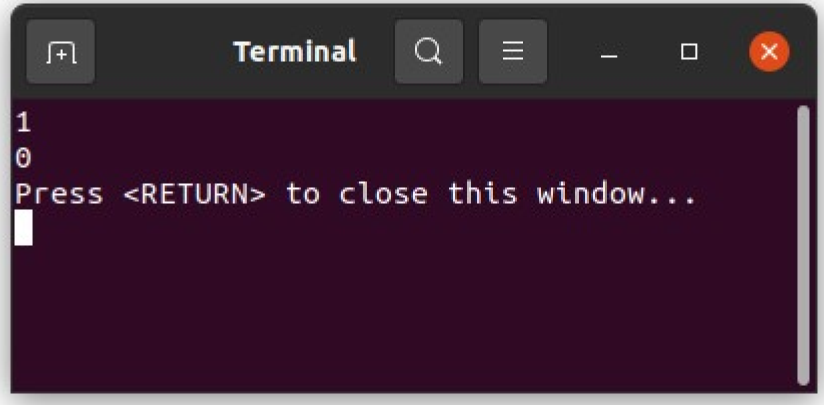
- A **Boolean** data type is declared with the **bool** keyword and can only take the values **true** or **false**.
- When the value is returned, **true = 1** and **false = 0**.

Boolean values are mostly used for conditional testing, which you will learn more about in a later chapter.

Example



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

int  main()
{
    bool  isCodingFun = true;
    bool  isFishTasty = false;

    cout << isCodingFun << endl;      // Outputs 1 (true)
    cout << isFishTasty << endl;       // Outputs 0 (false)

    return 0;
}
```

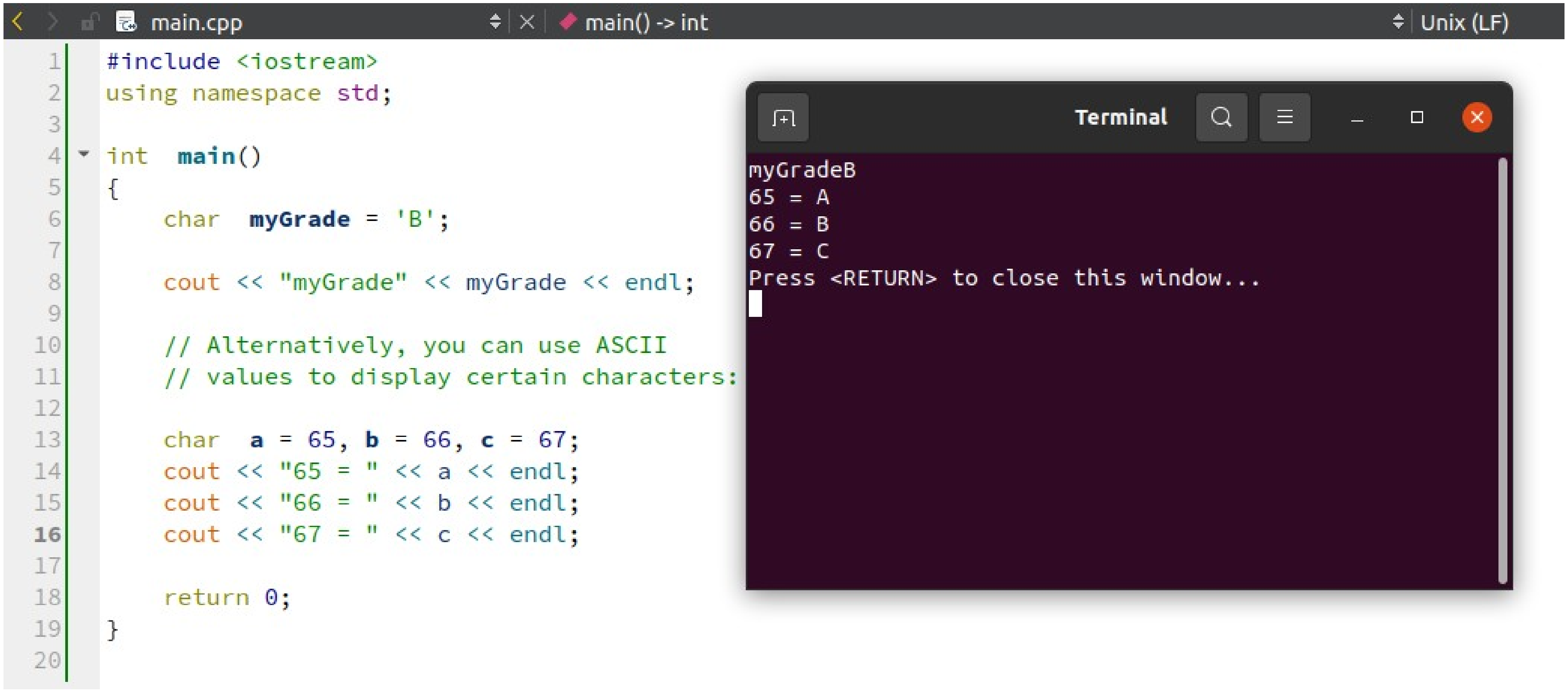


```
1
0
Press <RETURN> to close this window...
```

Character Types

- The **char** data type is used to store a single **character**. The character must be surrounded by single quotes, like **'A'** or **'c'**

Example



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

int  main()
{
    char  myGrade = 'B';

    cout << "myGrade" << myGrade << endl;

    // Alternatively, you can use ASCII
    // values to display certain characters:

    char  a = 65, b = 66, c = 67;
    cout << "65 = " << a << endl;
    cout << "66 = " << b << endl;
    cout << "67 = " << c << endl;

    return 0;
}
```

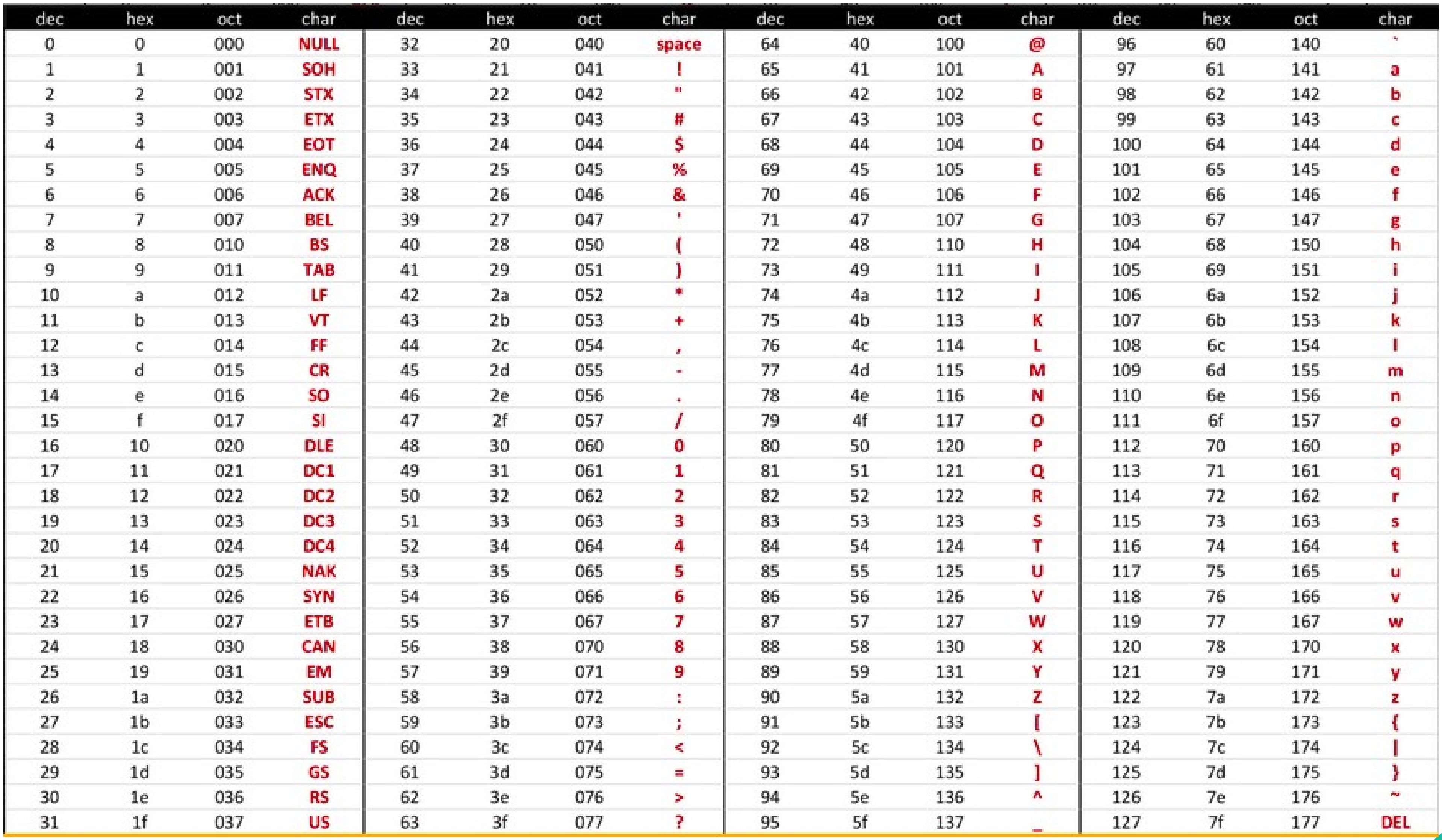
```
myGradeB
65 = A
66 = B
67 = C
Press <RETURN> to close this window...
```

ASCII

- The American Standard Code for Information Interchange (ASCII) is a means of encoding characters for digital communications. It was originally developed in the early 1960s as early networked communications were being developed.

Pars Pajouhaan
پارس پژوهان
بامجوز رسمی از سازمان فنی وحرفه‌ای

ASCII



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

String Types

- The string type is used to store a sequence of characters (text). This is not a built-in type, but it behaves like one in its most basic usage. String values must be surrounded by double quotes:

```
string greeting = "Hello";
cout << greeting;
```

پارس پژوهان

بامجوز رسمی از سازمان فنی وحرفه‌ای

Identifiers

- Identifiers == شناسه

- All C++ variables must be identified with unique names.
- These unique names are called identifiers.
- Identifiers can be short names (like x and y) or more descriptive names (age, sum, totalVolume).

Note: It is recommended to use descriptive names in order to create understandable and maintainable code.

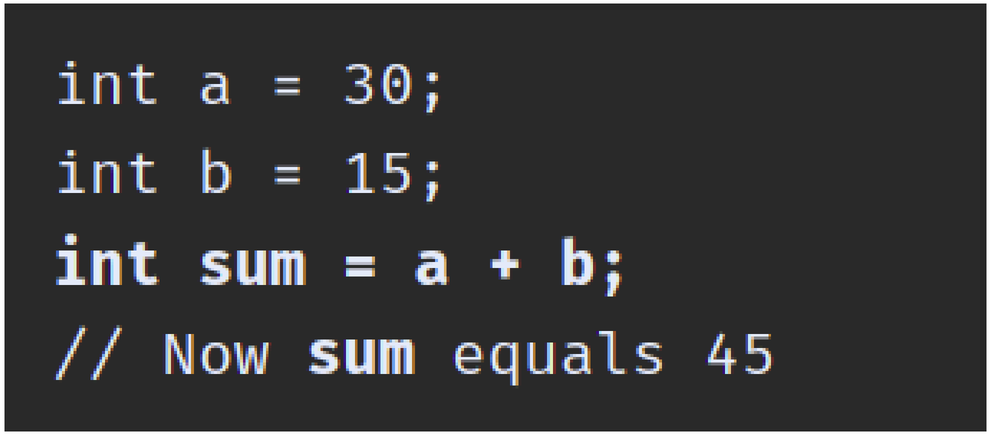
Identifiers

The general rules for naming variables are:

- Names can contain letters, digits and underscores
- Names must begin with a letter or an underscore (_)
- Names are case sensitive (`myVar` and `myvar` are different variables)
- Names cannot contain whitespaces or special characters like !, #, %, etc.
- Reserved words (like C++ keywords, such as `int`) cannot be used as names

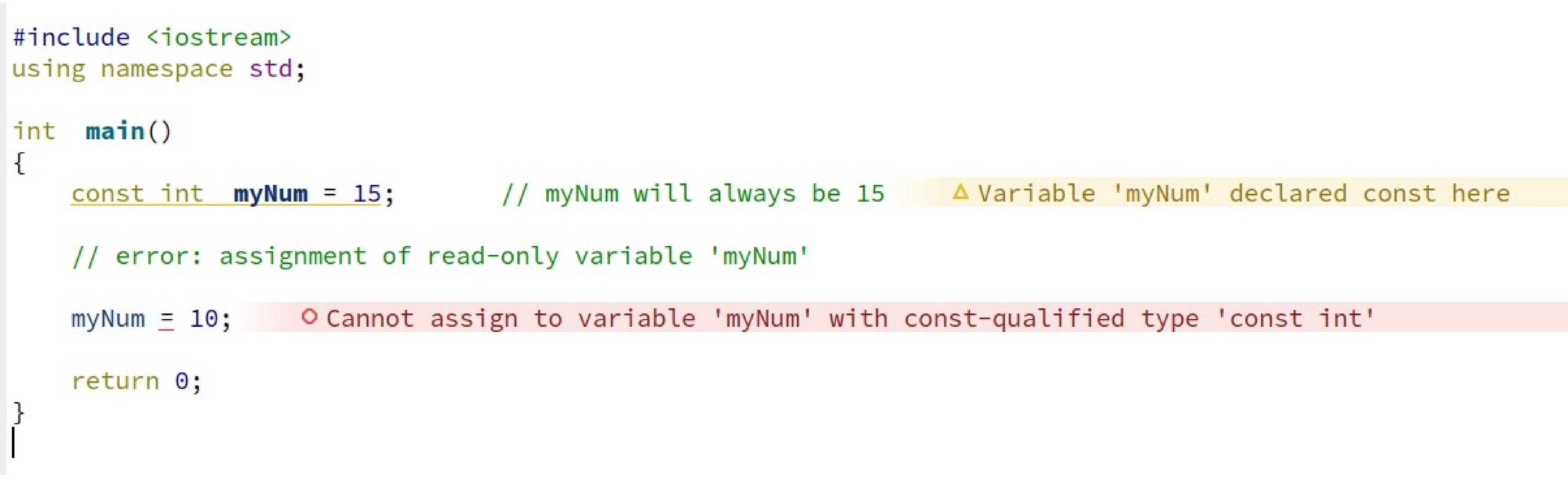
Variables

- A variable can be assigned a value, and can be used to perform operations.
- For example, we can create an additional variable called sum, and add two variables together.



```
int a = 30;
int b = 15;
int sum = a + b;
// Now sum equals 45
```

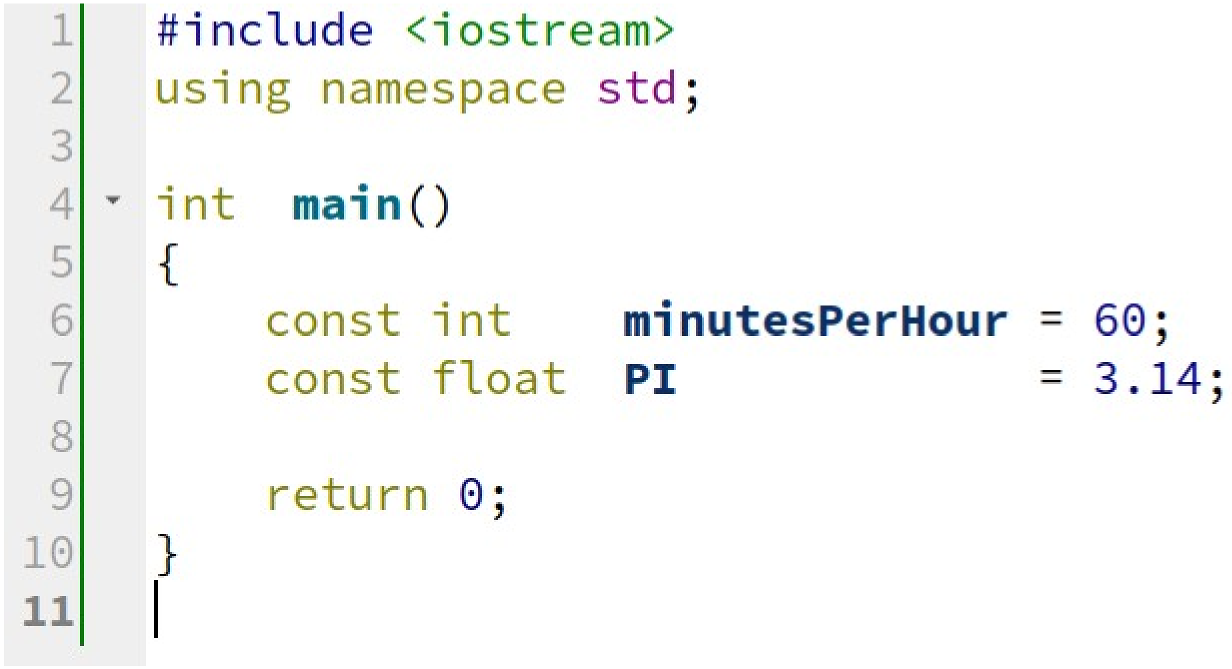
Constants



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

int  main()
{
    const int  myNum = 15;       // myNum will always be 15     ⚠ Variable 'myNum' declared const here

    // error: assignment of read-only variable 'myNum'

    myNum = 10;     ○ Cannot assign to variable 'myNum' with const-qualified type 'const int'

    return 0;
}
```

When you do not want others (or yourself) to **override** existing variable values, use the **const** keyword (this will declare the variable as "constant", which **means unchangeable and read-only**)

Constants



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

int  main()
{
    const int    minutesPerHour = 60;
    const float  PI             = 3.14;

    return 0;
}

```

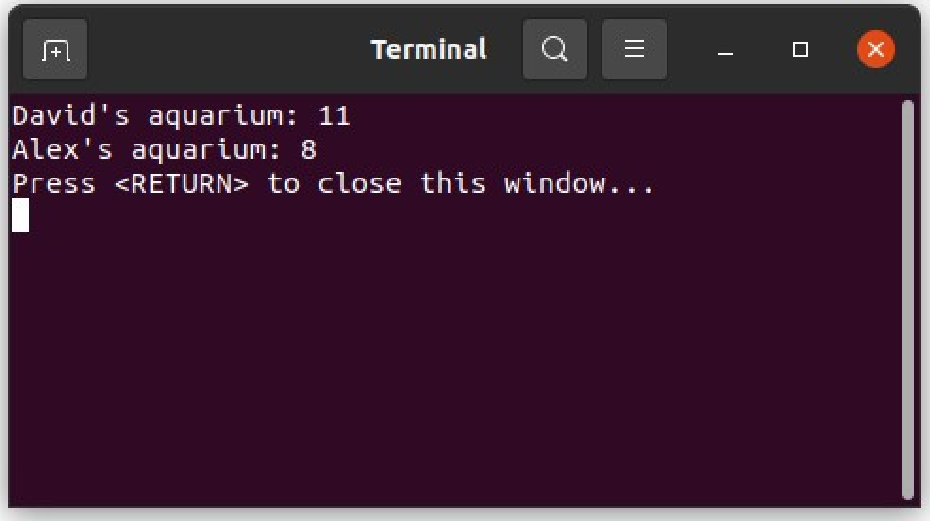
تمرین

David and Alex each have aquariums. There are 8 Rainbowfishes in David’s aquarium, and 11 Angelfishes in Alex’s aquarium. Help them exchange their fishes between them.

Pars Pajouhaan
پارس پژوهان
بامجوز رسمی از سازمان فنی و حرفه‌ای

پاسخ

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

int  main()
{
    int  aquariumDavid = 8;
    int  aquariumAlex  = 11;

// your code goes here
    int  s = aquariumDavid;

    aquariumDavid = aquariumAlex;

    aquariumAlex = s;

    cout << "David's aquarium: " << aquariumDavid << endl;
    cout << "Alex's aquarium: " << aquariumAlex << endl;

    return 0;
}
```



```
David's aquarium: 11
Alex's aquarium: 8
Press <RETURN> to close this window...
```

Question ?