Introduction

All variables use data-type during declaration to restrict the type of data to be stored. Therefore, we can say that data types tell the variables the type of data they can store.

Pre-defined data types available in C++ are:

- **int**: Integer value
- **unsigned int:** Can store only positive integers.
- **float**, **double**: Decimal number
- **char**: Character values (including special characters)
- **unsigned char**: Character values
- **bool**: Boolean values (true or false)
- **long**: Contains integer values but with the larger size
- **unsigned long**: Contains large positive integers or 0
- **short**: Contains integer values but with smaller size

Table for datatype and its size in C++: (This can vary from compiler to compiler and system to system depending on the version you are using)

asm	else	new	this
auto	enum	operator	throw
bool	explicit	private	true
break	export	protected	try
case	extern	public	typedef
catch	false	register	typeid
char	float	reinterpret_cast	typename
class	for	return	union
const	friend	short	unsigned
const_cast	goto	signed	using
continue	if	sizeof	virtual
default	inline	static	void
delete	int	static_cast	volatile
do	long	struct	wchar_t
double	mutable	switch	while
dynamic_cast	namespace	template	

Examples:

```
int price = 5000;                  // Integer (whole number)

float interestRate = 5.99f;        // Floating point number

char myLetter = 'D';               // Character

bool isPossible = true;            // Boolean

string myText = "Coding Ninjas";   // String
```

auto keyword in c++

The auto keyword specifies that the type of the declared variable will automatically be deduced from its initializer. It would set the variable type to initialize that variable's value type or set the function return type as the value to be returned.

Example:

```
auto a = 11          // will set the variable a as int type

auto b = 7.65        //will set the variable b as float

auto c = "abcdefg"   // will set the variable c as string
```