High School Programming

Lecture: 04

WELCOME TO

Mahinur Rahaman Hridoy

BSc in CSE, Diploma
Software Developer

Recap Previous Lecture

- C# Introduction
- Types of application
- Program structure, syntax, and comments
- Datatypes
- Variables

Agenda

- Keywords
- Variables and Constant
- Type Casting / Type Conversion
- User Inputs
- Outputs/Results

Keywords

-- C# contains reserved words that have special meaning for the compiler. These reserved words are called "**keywords**". Keywords cannot be used as an identifier (name of a variable, class, interface, etc.).

abstract	do	in	protected	throw
as	double	int	public	true
base	else	interface	readonly	try
bool	enum	internal	ref	typeof
break	event	is	return	unit
byte	explicit	lock	sbyte	ulong
case	extern	long	sealed	unchecked
catch	false	namespace	short	unsafe
char	finally	new	sizeof	ushort
checked	fixed	null	stackalloc	using
class	float	object	static	using static
const	for	operator	string	virtual
continue	foreach	out	struct	void
decimal	goto	override	switch	volatile
default	if	params	this	while
delegate	implicit	private		

Fig: C# Keywords

Variables and Constants

-- Variables are containers for storing data values. To create a variable, you must specify the type and assign it a value

Syntax:

datatype variableName = value;

const **datatype** variableName = value;

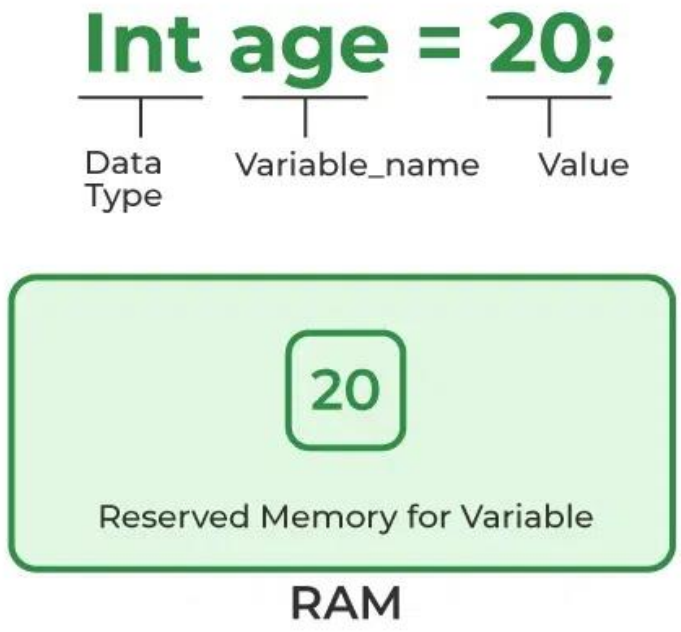

Fig: Variable

Type Casting / Type Conversion

Type casting is when you assign a value of one data type to another type.

In C#, there are two types of casting:

Implicit Casting (automatically) - converting a smaller type to a larger type size
char -> **int** -> **long** -> **float** -> **double**

Explicit Casting (manually) - converting a larger type to a smaller size type
double -> **float** -> **long** -> **int** -> **char**

User Inputs

-- In C#, the simplest method to get input from the user is by using the ReadLine() method of the Console class. However, Read() and ReadKey() are also available for getting input from the user. They are also included in Console class.

ReadLine(): The ReadLine() method reads the next line of input from the standard input stream. It returns the same string.

Read(): The **Read**() method reads the next character from the standard input stream. It returns the ascii value of the character.

ReadKey(): The **ReadKey**() method obtains the next key pressed by user. This method is usually used to hold the screen until user press a key.

Outputs || Results

-- To output values or print text in C#, you can print or shows any data as outputs.

WriteLine(): You can add as many WriteLine() methods as you want. Note that it will add a new line for each method.

Write(): There is also a Write() method, which is similar to WriteLine(). The only difference is that it does not insert a new line at the end of the output

Contact Me

Phone
+8801321869515

Website
www.mrhridoy.me

Mail
mrhridoy.me@gmail.com

Facebook
Mahinur Rahaman Hridoy

Thank You