

# Data Types and Variables

Introduction to Programming

## Introduction

- A computer program is a set of instructions for a computer to follow.
- Computers are used to process and store data.
- In order to store and process the data, the computer must be given instructions on:
  - The nature (or type) of the data
  - How the data is to be stored

## Data types

- The term **data type** refers to the nature of data – the types of values.
- A **data type** tells the computer how the data is to be represented in the program.
- This allows the computer to:
  - Allocate storage space for the data
  - Perform operations correctly

## Primitive Data Types in Java

Java type	Allows for
byte	Very small integers
short	Small integers
int	Big integers
long	Very big integers
float	Real numbers
double	Very big real numbers
char	Characters
boolean	True or false

## Primitive Data Types in Java

Java type	Allows for	Size
<b>byte</b>	very small integers	1 byte
<b>short</b>	small integers	2 bytes
<b>int</b>	big integers	4 bytes
<b>long</b>	very big integers	8 bytes
<b>float</b>	real numbers	4 bytes
<b>double</b>	very big real numbers	8 bytes
<b>char</b>	characters	2 bytes
<b>boolean</b>	true or false	1 bit

## Primitive Data Types in Java

Java type	Allows for	Range of values
<b>byte</b>	very small integers	-128 to 127
<b>short</b>	small integers	-32768 to 32767
<b>int</b>	big integers	-2147483648 to 2147483647
<b>long</b>	very big integers	-9223372036854775808 to 9223372036854775807
<b>float</b>	real numbers	+/- $1.4 * 10^{-45}$ to $3.4 * 10^{38}$
<b>double</b>	very big real numbers	+/- $4.9 * 10^{-324}$ to $1.8 * 10^{308}$
<b>char</b>	characters	Unicode character set
<b>boolean</b>	true or false	not applicable

## Integer data types

- An integer is a *positive* or *negative* whole number.  
E.g. 5, -22, 0
- An integer cannot contain a decimal point.
- Integers are represented in Java by the **byte**, **short**, **int** and **long** data types.
- During first year programming, we generally use the **int** data type for whole numbers.

## Real number data types

- A real number is a positive or negative number with a decimal point. E.g 5.5, -25.0, 0.0
- Real numbers are represented in Java by the **float** and **double** data types.
- During first year programming, we generally use the **double** data type.

# Variables

- A **variable** is a named storage location in the computer's memory.
- This named location or **variable** is used to store data while the program is running.
- **Variables** contain values that the program can manipulate.
- The value of a **variable** can be changed (or can vary) during execution of the program.

## Declaring Variables

- To create a stored memory location you must **declare** a variable
- This declaration states the type of data the variable is to contain and gives the variable a name
- A **variable** is required for every piece of information that your program will store

# Declaring Variables

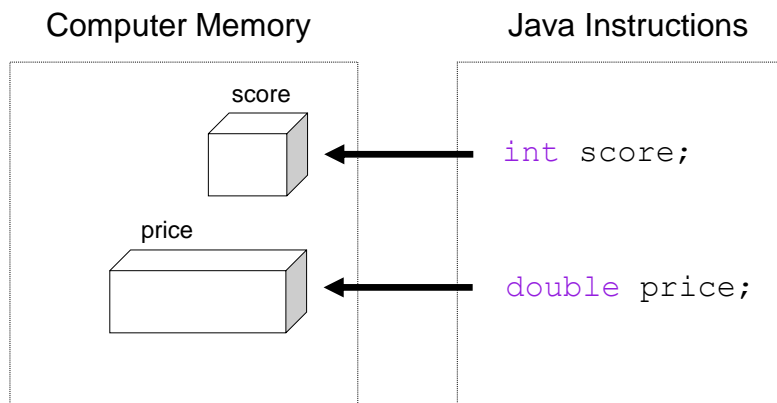
## Syntax:

```
dataType variableName;
```

## Example:

```
int age;  
double pricePerUnit;
```

# Declaring Variables



## Rules for naming Variables

1. Can use letters, numerals or underscores and dollar (A-Z a-z 0-9 \_ and \$) (but don't use \$)
2. The first character of the name must be a letter, an underscore, (or a dollar sign)
  - Cannot start with a digit
3. The name cannot be a *keyword* in the Java language
4. The name cannot contain spaces
5. The name cannot contain any symbols such as ! % @ etc.

## Java Reserved Words

abstract	do	if	package	synchronized
boolean	double	implements	private	this
break	else	import	protected	throw
byte	extends	instanceof	public	throws
case	false	int	return	transient
catch	final	interface	short	true
char	finally	long	static	try
class	float	native	strictfp	void
const	for	new	super	volatile
continue	goto	null	switch	while
default				

- Note that `const` and `goto` are reserved, but they are not used.
- `true`, `false`, and `null` are actually literals.

## Good Programming Practice

- Use meaningful variable names
  - makes a program self documenting i.e. fewer comments needed
- The first letter of variable name/identifier should be lowercase.
- Multiple word variable names can help make a program more readable.
  - But avoid running separate words together e.g.  
`totalmonthlysales`
  - Separate the words with underscores e.g.  
`total_monthly_sales`
  - or begin each word after the first with a capital letter e.g.  
`totalMonthlySales` (\*\*use this one\*\*)

## Naming Variables

- Variable names should start with a lower-case letter
- Java is case sensitive – the compiler will recognise `firstNumber` and `firstnumber` as different variables.
- It is good programming practice to give meaningful names to your variables.



## Variable assignment

- Use the **assignment operator** (=) to assign a value to a variable

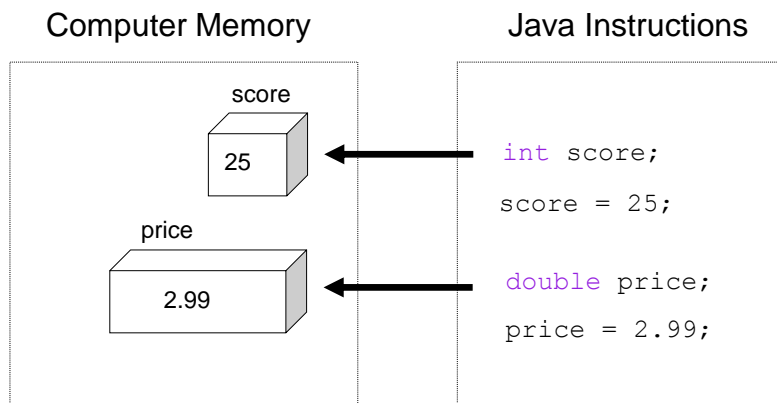
- E.g.

```
int balance;  
balance = 100;  
balance = 100 + 20;
```

- Alternatively:

```
int balance = 120;
```

## Variable assignment



## Displaying Variable values

- `System.out.print()` or `System.out.println()` can be used to display the values contained in variables.

```
int myAge = 21;  
System.out.print(myAge);
```

- Note the absence of double quotes.

## Character Variables

- Character variables hold a single character e.g. 'A'
- Use the `char` keyword to declare a character variable

```
char grade;
```

- Use single quotes when assigning a value to a character variable

```
grade = 'B';
```