

# Getting Started with Java

## Variables

# What is a Variable?

A useful program needs to modify, store and retrieve information. The areas in computer memory where data is stored is called a variable.

From our discussion of data types, we discovered that there are many specific types of data. One or many variables can be created for each type.

Variables are given names to help programmers remember and understand their data type and their purpose.

# Naming Variables

The name given to a variable is called the identifier. There are some rules and conventions associated with naming variables.

**Rules:** Java requires that you follow certain rules when naming variables. If you break these rules, your program probably won't even compile.

**Conventions:** Programmers have agreed that certain rules for naming make better code. These make your code, and probably mark, better.

# Naming Variables – Rules

- any character used in the name must be
  - a letter of the alphabet (a to z, A to Z), or
  - a digit from 0 to 9, or
  - the underscore character \_
- the first character cannot be a digit (0 to 9)
- Java has certain reserved words that have a particular meaning to the Java language, which cannot be used

# Naming Variables – Conventions

- PascalCase
  - each word begins with a capital letter
  - **applies to classes**
- camelCase
  - the first letter in the name is lower case
  - any other words begin with a capital letter
  - **applies to variables and methods**

# Declaring Variables

- reserves space in memory for the variable
- specify the type of data to be stored
- name using camelCase

```
int count;
```

```
float area;
```

```
boolean isDoorClosed;
```

# Assigning Values to Variables

- declaring a variable creates space in memory, but the space is empty
  - the variable is said to be undefined or uninitialized
- there are several ways to assign a value to a variable
  - when the variable is declared (initialize)
  - immediately after declaration (initialize)
  - later in the program (assigning a value)

# Initializing a Variable

- as soon as a variable is declared, it is good practice to set a value
- using an uninitialized variable will produce an error, or unexpected behaviour

```
int count = 0;    // start count at 0
```

```
float area;
```

```
area = -1.0;     // area cannot be neg
```



# Assigning a Value to a Variable

- variables are used to store data as the program executes
- the values may change based on user input or calculations made by the program

```
int count = 0;
```

```
int numStudents = 20;
```

```
...
```

```
count = numStudents;
```

# Modifying a Variable

- sometimes, the value in a variable needs to be changed with respect to the current value of the variable
  - for example, increasing or decreasing by some amount
  - an increase by 1 is called incrementing, and a decrease by 1 is called decrementing

```
int count = 0;           // count is now 0
```

```
count = count + 1;      // count is now 1
```

```
count = count + 1;      // count is now 2
```