

# Variables and Types



# Overview

- One role of a computer program is to model and manipulate real or imaginary world entities
- In this module, we will learn how to represent the properties (or attributes) of the entities that our program will manipulate
- Example:
  - Entity: student
  - Properties: name, age, height, etc.

# Learning Objectives

- To know what a variable is
- To be able to declare a variables
- To be familiar with primitive types
- To be able to perform operations on primitive types variables
- To be able to solve problems using primitive types variables

# Definitions

- A **variable** helps us capture and store details about the problem or entity we are solving or modeling
- A **variable** is a **name** that is associated with a **value (data)**
- The **value** is stored in a **memory** location in the computer
- The **value** can **change**
- A **variable** (value) must have a **type**
- A **type** is a group of values and the operations that can be performed on them

# Modeling with variables

- We are building a program to keep track of the **CIS 110 students**; we need to record information about them
- What information should we store?

Information / variable	Examples	Type
Name	Malcom, Maya, Toni,...	Text
Age	13, 15, ...	Number
Is a CIS major?	True , False	Text
Height	5.7. 6.0, 4.2, ...	Number

- Can you think of any additional information that we could store?

# Data types

- Two types of data types
  - Primitive types
  - Object types

# Primitive types

- `int`: stores whole numbers (positive or negative) like 3, -5, 19000
- `double`: stores decimal numbers (positive or negative) like 3.5, -5.1, 19000.1
- `boolean`: stores Boolean values, either `true` or `false`

# Object types

- They hold a reference to an object
- **String** is an object type and is associated with text (sequence of characters) values



# Modeling with variables and Java types

- We are building a program to keep track of the **CIS 110 students**; we need to record information about them
- We update our table to use Java types

Information / variable	Examples	Java Type
Name	Malcom, Maya, Toni,...	<b>String</b>
Age	17, 15, ...	<b>int</b>
Is a CIS major?	True , False	<b>boolean</b>
Height	5.7. 6.0, 4.2, ...	<b>double</b>

- What is the Java type of the information you added?

# Variable declaration

- Creates a variable
- Associates a variable to a type
- The type determines how many bits (number of 0s and 1s) the computer will use to store the value associated with the variable
- Examples

```
// declaring the variable name
```

```
String name;
```

```
// declaring the variable age
```

```
int age;
```

# Variable initialization

- Assigns a value to a variable: using the = sign
- The value and the type of the variable must be compatible
- Examples

// declaring and initializing the variable name

```
String name = "Malcom";
```

variable ← value

Always surround a String value with "

// declaring the variable age

```
int age;
```

```
age = 14;
```

variable ← value

One line: declaration + initialization

Two steps: declaration then initialization

```
boolean is_taking_CIS110 = true;
```

# Print variable

- Put the variable name without the quotes in the print command

```
String name = "Maya";  
System.out.print(name);  
Prints Maya
```

- Use the **+** operator to append the value of a variable to a text in the print command

```
System.out.print("Name of the student: " + name);  
Prints Name of the student: Maya
```

# Operations on variables

- Assignment statement initializes or changes the value of a variable previously declared
- Mathematical operators apply to `int` and `double` values

# Operations on int

Type of operand 1	Operator	Type of operand 2	Type of result	Example
int	+	int	int	$3 + 5 == 8$
int	-	int	int	$4 - 6 == -2$
int	*	int	int	$2 * 3 == 6$
int	/	int	<b>int</b>	$3 / 2 == 1$

# Operations on double

Type of operand 1	Operator	Type of operand 2	Type of result	Example
double	+	double	double	$3.5 + 5.5 == 9.0$
double	-	double	double	$4.0 - 6.0 == -2.0$
double	*	double	double	$2.5 * 1.0 == 2.5$
double	/	double	double	$3.0 / 2.0 == 1.5$

# Operations on double and int

- When one of the operand is of type double, the results is of type double

Type of operand 1	Operator	Type of operand 2	Type of result	Example
double	+	int	double	$3.5 + 5 == 8.5$
int	-	double	double	$4 - 6.0 == -2.0$
double	*	int	double	$2.5 * 1 == 2.5$
double	/	int	double	$3.0 / 2 == 1.5$



# Equality

- The `==` operator is used to check for equality.
- The result is a `boolean` value (`true` or `false`)

```
int x = 4;  
int y = 5;  
x == y; // prints false
```

- The result of the comparison can be stored in a `boolean` value

```
int x = 4;  
int y = 5;  
boolean v = (x == y);  
System.out.print(v); // prints false
```

# Inequality

- The `!=` operator is used to check for inequality (not equals).
- The result is a `boolean` value (`true` or `false`)

```
int x = 4;
```

```
int y = 5;
```

```
x != y; // prints true
```

- The result of the comparison can be stored in a `boolean` value

```
int x = 4;
```

```
int y = 5;
```

```
boolean v = (x != y);
```

```
System.out.print(v); // prints true
```

# The modulo (%) operator

- The mod operator ( $x \% y$ ) returns the remainder after you divide  $x$  (first number) by  $y$  (second number)

$5 \% 2 \rightarrow 1$

$4 \% 2 \rightarrow 0$

# Compound Assignment Operators

- Shortcuts that do a math operation and assignment in one step

<b>+ shortcuts</b>	<b>- shortcuts</b>	<b>* shortcut</b>	<b>/ shortcut</b>	<b>% shortcut</b>
<code>x = x + 1;</code>	<code>x = x - 1;</code>	<code>x = x * 2;</code>	<code>x = x / 2;</code>	<code>x = x % 2;</code>
<code>x += 1;</code>	<code>x -= 1;</code>	<code>x *= 2;</code>	<code>x /= 2;</code>	<code>x %= 2;</code>
<code>x++;</code>	<code>x--;</code>			

# Casting

- Type casting converts a variable from one type to another

- `(int)` is used to cast a value to `int`

`(int) 3.5` → 3

- `(double)` is used to cast a value to `double`

`(double) 3/2` → 1.5

- Casting applies only to the closest operand