Primitive Types

Variables: *Declaring*, *Initializing*, *Assigning*

Lecture Contents

- ***Declaring*** and ***Initializing*** variables
- Identifier Conventions
- Reserved Words
- ***Assigning*** variables

Declaring and Initializing Variables

- A ***variable*** is a memory location to store information/data
- An ***identifier*** is a name given to something so it can be referenced.
- ***Declaring*** a *variable* is telling the compiler that you will use a chosen *identifier* to refer to some particular type of data.
- When a *variable* is ***initialized***, it is *assigned* a value for the first time.
- *Variables* may be ***assigned*** new values after *initialization.*
- *Variables* are *declared* and *initialized* with:
  - ***<type> <identifier> = <value> ;***

Declaring and Initializing Variables

- ***Variables*** are *declared* and *initialized* with:
  - **<type> <identifier> = <value> ;**

```
int myInteger = 13;

double myReal = 1.78;

boolean isTrue = false;

String myText = “Hello World!”;
```

Reserved Words

- These words cannot be used as labels:

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

Declaring and Initializing Variables

- ***Variables*** can be *declared* and *initialized* separately:
  - **<type> <identifier> = <value> ;**

```
int myInteger = 13;
```

```
int myInteger;
myInteger = 13;
```

```
double myReal = 1.78;
```

```
double myReal;
myReal = 1.78;
```

```
boolean isTrue = false;
```

```
boolean isTrue = false;
isTrue = false;
```

```
String myText = “Hello World!”;
```

```
String myText;
myText = “Hello World!”;
```

Identifier Conventions

- Class: PascalCase (UpperCamelCase)
- Variables/Methods: camelCase (lowerCamelCase)
- Final Variables: UPPER_CASE_WITH_UNDERSCORES

Examples

- Variables must be *declared* and *initialized* before they are used.

```
public static void main(String[] args) {
    int myNumber;
    System.out.println(myNumber);
    myNumber = 13;
}
```

Examples

- Variables must be *declared* and *initialized* before they are used.

```
public static void main(String[] args) {
    int myNumber;
    System.out.println(myNumber);
    myNumber = 13;
}
```

```
ERROR:
The local variable myNumber may not have
been initialized
```

Variable *Assignment*

- Initialization is an assignment
- Variables can be assigned new values after being initialized

```
public static void main(String[] args) {
    int myNumber = 13;
    myNumber = 7;
    System.out.println(myNumber);
}
```

Declaring Variables

```
public static void main(String[] args) {
    int myNumber = 13;
    myNumber = 7;
    System.out.println(myNumber);
}
```

```
7
```

Declaring Variables

```
public static void main(String[] args) {
    int myNumber = 13;
    myNumber = myNumber + 7;
    System.out.println(myNumber);
}
```

Declaring Variables

```
public static void main(String[] args) {
    int myNumber = 13;
    myNumber = myNumber + 7;
    System.out.println(myNumber);
}
```

```
20
```

Declaring Variables

```
public static void main(String[] args) {
    int myNumber = 13;
    double myFloat = 6.5;
    System.out.println(myNumber/myFloat);
}
```

Declaring Variables

```
public static void main(String[] args) {
    int myNumber = 13;
    double myFloat = 6.5;
    System.out.println(myNumber/myFloat);
}
```

```
2.0
```

Declaring Variables

```
public static void main(String[] args) {
    int myNumber = 13;
    double myFloat = 2.5;
    myFloat = myNumber/myFloat;
    System.out.println(myFloat);
}
```

Declaring Variables

```
public static void main(String[] args) {
    int myNumber = 13;
    double myFloat = 2.5;
    myFloat = myNumber/myFloat;
    System.out.println(myFloat);
}
```

```
5.2
```

Declaring Variables

```
public static void main(String[] args) {
    int myNumber = 13;
    double myFloat = 2.5;
    myNumber = myNumber/myFloat;
    System.out.println(myNumber);
}
```

Declaring Variables

```
public static void main(String[] args) {
    int myNumber = 13;
    double myFloat = 2.5;
    myNumber = myNumber/myFloat;
    System.out.println(myNumber);
}
```

```
ERROR:
Type mismatch: cannot convert from
double to int
```

Declaring Variables

```
public static void main(String[] args) {
    int myNumber = 13;
    double myFloat = 2.5;
    myNumber = (int)(myNumber/myFloat);
    System.out.println(myNumber);
}
```

Declaring Variables

```
public static void main(String[] args) {
    int myNumber = 13;
    double myFloat = 2.5;
    myNumber = (int)(myNumber/myFloat);
    System.out.println(myNumber);
}
```

```
5
```

Declaring Variables

```
public static void main(String[] args) {
    int myNumber = 13;
    double myFloat = 2.5;
    myNumber = myNumber/(int)myFloat;
    System.out.println(myNumber);
}
```

Declaring Variables

```
public static void main(String[] args) {
    int myNumber = 13;
    double myFloat = 2.5;
    myNumber = myNumber/(int)myFloat;
    System.out.println(myNumber);
}
```

```
6
```

Declaring Variables

```
public static void main(String[] args) {
    int myNumber = 13;
    double myFloat = 2.5;
    myNumber = (int)myNumber/myFloat;
    System.out.println(myNumber);
}
```

Declaring Variables

```
public static void main(String[] args) {
    int myNumber = 13;
    double myFloat = 2.5;
    myNumber = (int)myNumber/myFloat;
    System.out.println(myNumber);
}
```

```
ERROR:
Type mismatch: cannot convert from
double to int
```

Declaring Variables

- ***Variables*** are declared with:
  - **<type> <identifier> = <value> ;**

```
int myInteger = 13;

double myReal = 1.78;

boolean isTrue = false;

String myText = “Hello World!”;
```

Java Primitive Types

- Integers
  - byte (1)
  - short (2)
  - **int** (4)
  - long (8)
- Real
  - float (4)
  - **double** (8)
- True/False
  - **boolean** (1 bit?)
- Letters
  - **char** (2)

Unless told otherwise, for assignments just use the primitive types that are given in **bold** font.

Primitive Types

Integers and Doubles