## Debugger

Debugger is a program that facilitates debugging. You can use a debugger to:

Execute a single statement at a time.

Trace into or stepping over a method.

Set breakpoints.

Display variables.

Display call stack.

Modify variables

## Quiz 1

1. In Java a class called Book should be stored in a file called Book.java

2. The name of a Java class must begin with a capital letter, otherwise the program will not run

3. Single line comments using // can be nested inside block comments using /* */

4. Each opening brace { must have a corresponding closing brace }

5. Java methods are private by default, which means they cannot be called from outside of that file.

6. If you want to use Eclipse to develop your Java programs, your compute must also have a Java Development Kit (JDK) installed

write a program to print your name.

## Variables

In a program, the variables store data.

Primitive variables store single pieces of data

(ex: char): char letter = 'A';

Object or reference type variables store multiple pieces of data

(ex: a String is a sequence of potentially multiple characters):

String text = "ABCDEFG";

## Variables

All Java variables must have a declared type

A variable's type determines:

what kind of value the variable can hold

how much memory to reserve for that variable

char letter;

int i;

double area;

String s;

Object o;

## Java's Primitive Types

Integers (whole numbers)

 byte–1 byte (-128 to 127)

 short –2 bytes (-32768 to 32767)

 int–4 bytes (-2147483648 to 2147483647)

 long–8 bytes (-9223372036854775808 to 9223372036854775807)

Real Numbers

 float–4 bytes

 double–8 bytes

char–2 bytes  stores a single character (Unicode 2)

boolean–stores true or false (uses 1-bit or byte)

## Variables

A variable must be declared before being assigned values.

Variables can be declared and initialized at once.

A variable must be initialized before being referenced.

A variable should only be declared once.

Variables can only be used inside the block { …} or scope that they themselves are declared.

**claim.Academy**
{st. louis}

## Variables

The Assignment Statement

    variable = expression;

What does it do?

    Solves/evaluates expression first!

    Assigns resulting value to the variable!

Exercise: What's the output?

```java
int x = 5;
x = x + x + x + 10;
System.out.print(x);
```

## Variables

Assignment Compatibility:

The variable and expression should be the same type

if not, you may get a compiler error.

Examples:

```
int sumGrades, gradeX, gradeY;

gradeX = 1;

sumGrades = 1473;

sumGrades = 1472 + 1;

sumGrades = 1472 + gradeX;

sumGrades = true; // ILLEGAL // COMPILER ERROR
```

## Variables

What about mixing numeric types?

Are these assignment statements ok?

int x = 5;

long y = x;

double z = y;

What about these?

double a = 6.5;

long b = a;

int c = b;

byte < short < int < long < float < double

No assigning big types to little types OR real types to integer types

## Variables

Type Casting as a type override

temporarily change a data type to another type

(type_name), example: (int)

no type casting is allowed to/from boolean

Examples:

double myReal = 10.0;

int badInt = myReal; // Error

int goodInt = (int)myReal;//Good

## Methods

A method is a collection of statements that performs a sequence of operations

It is used by invoking a statement with arguments

System.out.println("Welcome to Java!");

## The main Method

The main method provides the control of program flow.

ClassName is executable because it has a main method

    we can compile and then run it

Not all classes require main methods  only those classes that initiate program execution require a main method

```
public class ClassName {

    public static void main(String[] args) { … }

}
```

## Our first program: HelloWorldApp.java

```
/**
* HelloWorldApp is a Java application
* that simply displays "Hello World!" in the
* Java console.
*/
public class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
        // Statement above displays "Hello, World!"
    } }
```

1. Create a Scanner object

> Scanner input = new Scanner(System.in);

2. Use the methods next(), nextByte(), nextShort(), nextInt(), nextLong(), nextFloat(), nextDouble(), or nextBoolean() to obtain a String, byte, short, int, long, float, double, or boolean value.

For example,

System.out.print("Enter a double value: ");

> Scanner input = new Scanner(System.in);

> double d = input.nextDouble();

Scanner is in the Java package java.util

- start your program with:

> import java.util.Scanner;