UNIVERSITÉ Concordia UNIVERSITY

CENTRE FOR
CONTINUING EDUCATION

INTRODUCTION TO PROGRAMMING WITH JAVA - CEJV416

LECTURE #1

A brief history

The C Language

- C came into being in the years 1969-1973, in parallel with the early development of the Unix operating system

- C was developed to simplify system programming that was previously done in assembly language.

- In 1978 Brian Kernighan and Dennis Ritchie published *The C Programming Language*

A Brief Java History

- 1991 Sun Microsystems began the Green project
- Goal was to develop a programming language for intelligent consumer electronic devices such as cable TV set top boxes
- Language, first called Oak, was based on the C/C++ language
- Sun could not interest cable companies and was on the verge of canceling the project
- 1993 saw the World Wide Web explode in popularity and Sun saw an application of Oak in the area of dynamic web page creation

A Brief Java History

- Sun discovered that another company had the rights to the name Oak and so a new name was required
- James Gosling, lead developer and Canadian, came up with name Java while having coffee at a local coffee shop
- Sun announced the Java language in 1995 and immediately attracted attention due to its relationship with the WWW
- Today Java is used on cell phones, PDAs, PCs, and servers
- Google Android system is based on Java
- Read more about the history of Java in this article
  - http://www.wired.com/wired/archive/3.12/java.saga.html

Java Timeline – Java 8 released 2014/03/18

Year	Month	Release
1996	January	Java Development Kit 1.0 (JDK 1.0)
1997	February	Java Development Kit 1.1 (JDK 1.1)
1998	December	Java 2 Platform with version 1.2 of the Software Development Kit (SDK 1.2)
1999	August	Java 2 Platform, Standard Edition (J2SE)
	December	Java 2 Platform, Enterprise Edition (J2EE)
2000	May	J2SE with version 1.3 of the SDK
2002	February	J2SE with version 1.4 of the SDK
2004	September	J2SE with version 5.0 (instead of 1.5) of the JDK
2006	December	Java SE 6 with version 1.6 of the JDK
2011	July	Java SE 7 with version 1.7 of the JDK

Programming Methodologies I

spaghetti programming

An ancient (approximately 1950-1978) method of programming where programs were viewed as nothing more than sequences of instructions.

Programmers were required to think like computers.

All logical branching were made via "goto" statements, with the branched-to code not knowing what called it.

The logic flow of such programs looked like a bowl of spaghetti, hence the name.

Spaghetti programs were incredibly difficult to maintain, and couldn't be expanded much beyond a few thousand lines of code, because everything in the program depended on everything else.

Spaghetti Programming

```
10 REM Guess A Number Game

20 RANDOMIZE TIMER

30 SN = INT(RND * 10) + 1

40 INPUT "Guess a number
 from 1 to 10", MN

50 IF MN < SN THEN PRINT
 "Too small!": GOTO 40

60 IF MN > SN THEN PRINT
 "Too large!": GOTO 40

70 PRINT "You Win"

80 END
```

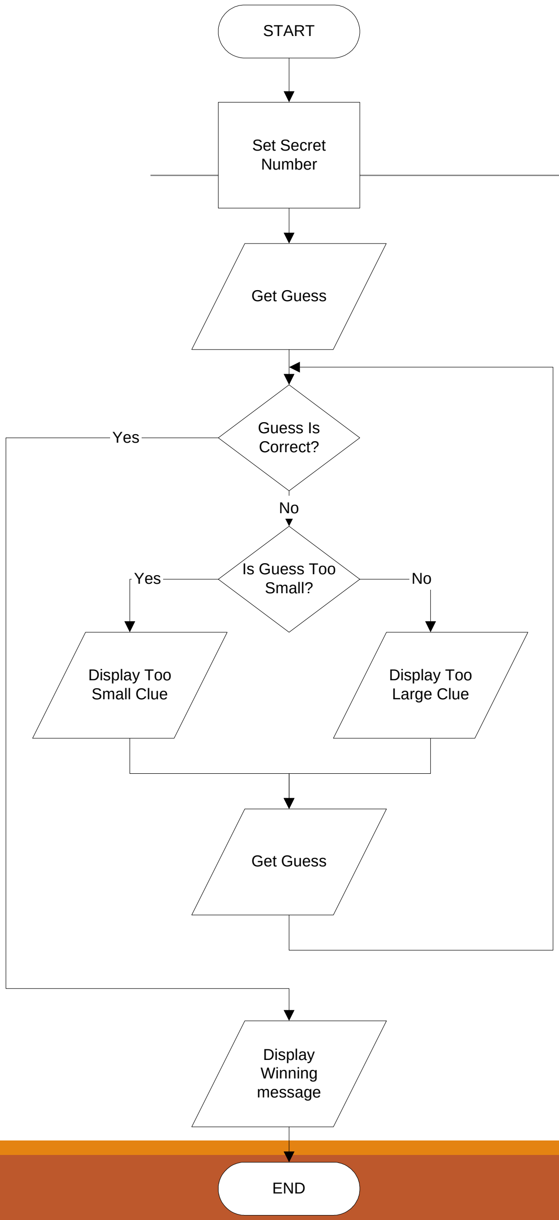
Programming Methodologies II

structured programming:

A technique for organizing and coding computer programs in which a hierarchy of modules is used

Each module has a single entry and a single exit point

Structured Programming



```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int setSecretNumber();
int getGuess();
void giveSmallClue();
void giveLargeClue();
void giveWinningMessage();

void main(void)
{
   int sn;
   int mn;
   sn = setSecretNumber();
   mn = getGuess();
   while (sn != mn)
   {
      if (sn > mn)
    giveSmallClue();
 else
    giveLargeClue();
 mn = getGuess();
   }
   giveWinningMessage();
}

int setSecretNumber(void)
{
   srand( (unsigned)time(NULL));
   return ((rand() % 10) + 1);
}
```

```
int getGuess(void)
{
   int localGuess;
   printf("Guess a number from 1-10: ");
   scanf("%d", &localGuess);
   return (localGuess);
}

void giveSmallClue(void)
{
   printf("Too small!\n");
}

void giveLargeClue(void)
{
   printf("Too large!\n");
}
void giveWinningMessage(void)
{
   printf("You win!\n");
}
```

Programming Methodologies III

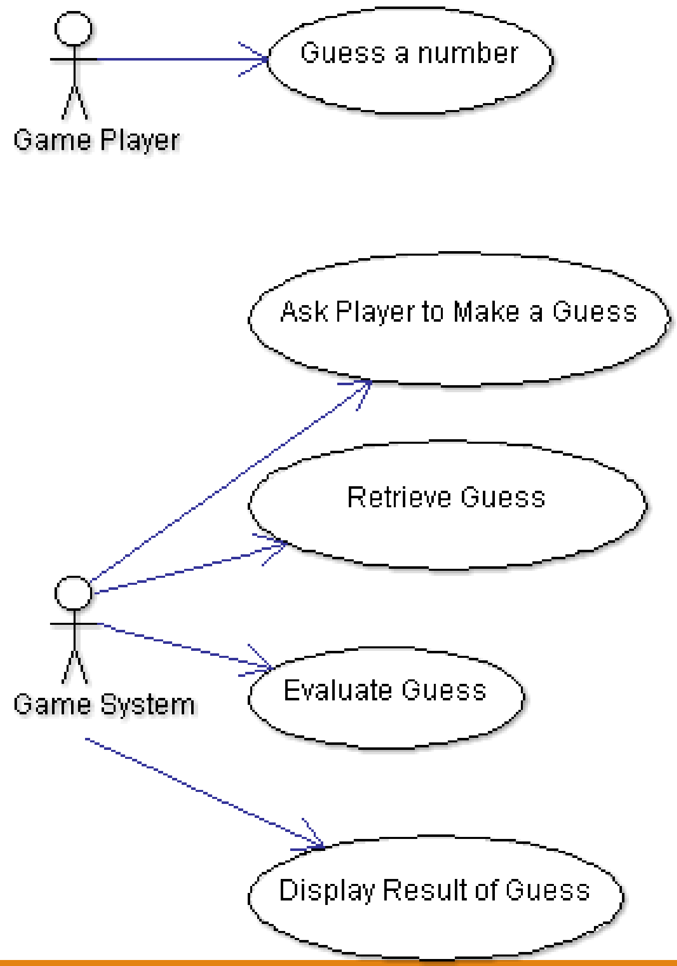
object-oriented programming

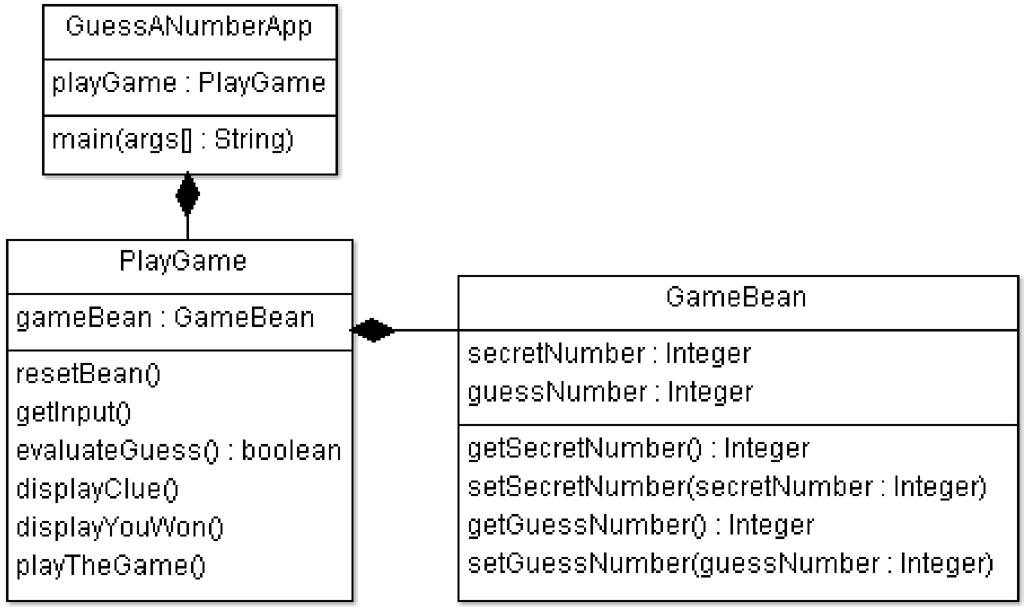
A type of programming in which programmers define not only the data type of a data structure, but also the types of operations that can be applied to the data structure.

The data structure becomes an *object* that includes both data and functions.

Relationships can be created between one object and another.

Object Oriented Programming





Object Oriented Programming

```
 package com.cejv416.game;

public class GameBean {

    private int secretNumber;
    private int guessNumber;

    public GameBean() {
        guessNumber = 0;
        secretNumber = (int)(Math.random() * 10 + 1);
    }

    public int getSecretNumber() {
        return secretNumber;
    }

    public void setSecretNumber(int secretNumber) {
        this.secretNumber = secretNumber;
    }

    public int getGuessNumber() {
        return guessNumber;
    }

    public void setGuessNumber(int guessNumber) {
        this.guessNumber = guessNumber;
    }
}
```

```
package com.cejv416.game;

public class GuessANumberApp {
    public static void main(String[] args) {

        PlayGame pg = new PlayGame();
        pg.playTheGame();
        System.exit(0);
    }
}
```

Object Oriented Programming

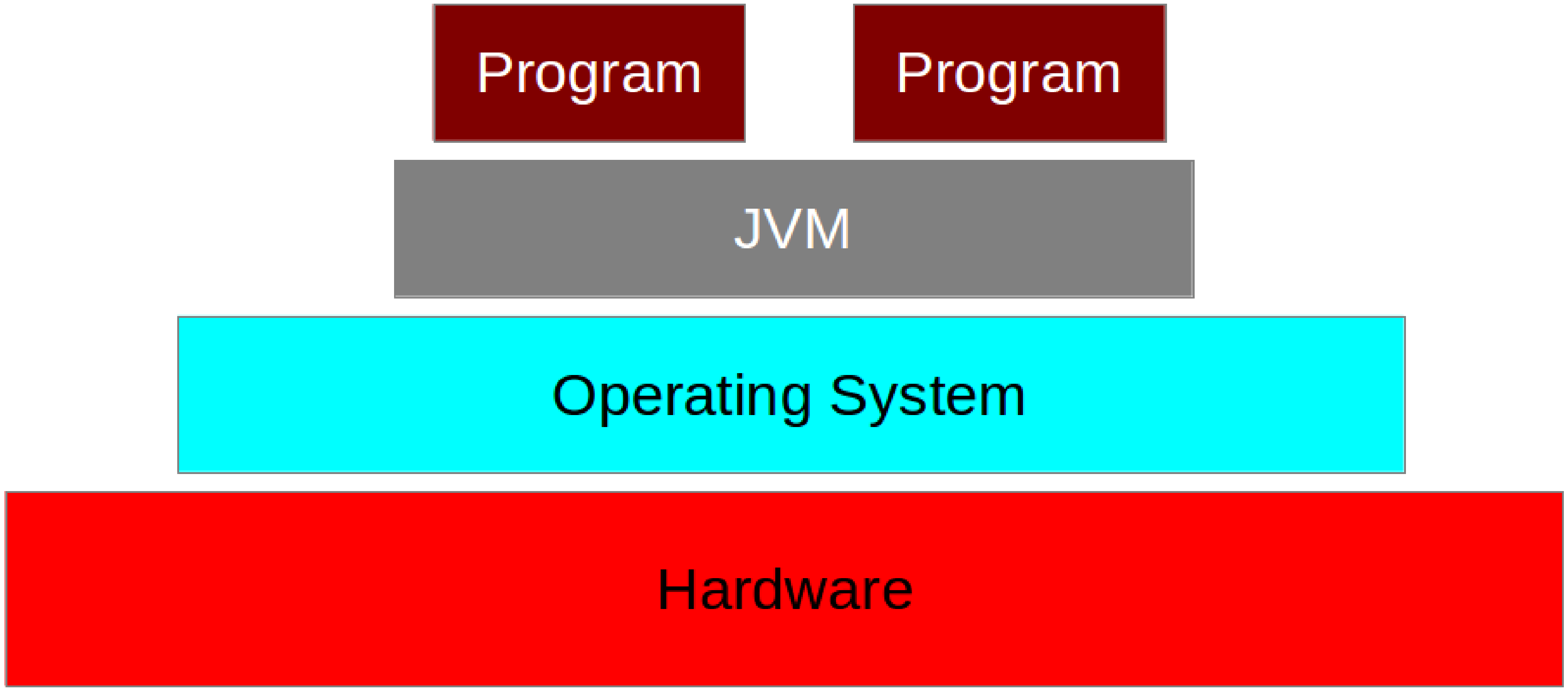
```
package com.cejv416.game;

import java.util.Scanner;

public class PlayGame {
    private GameBean gameBean;
    private Scanner sc;

    public PlayGame() {
        sc = new Scanner(System.in);
        resetBean();
    }
    private void resetBean() {
        gameBean = new GameBean();
    }
    private void getInput() {
        int guess = 0;
        guess = sc.nextInt();
        gameBean.setGuessNumber(guess);
    }
    private boolean evaluateGuess() {
        boolean retVal = false;
        if (gameBean.getSecretNumber()==
                   gameBean.getGuessNumber()) {
            retVal = true;
        }
        return retVal;
    }
private void displayClue() {
   if (gameBean.getSecretNumber() >
                    gameBean.getGuessNumber()) {
      System.out.println("Guess a larger number");
   } else {
      System.out.println("Guess a smaller number");
   }
}
private void displayYouWon() {
    for(int x = 0; x < 5; ++x) {
       System.out.println("YOU WON !!!");
    }
}
public void playTheGame() {
   boolean win = false;
   do {
      System.out.println("Guess a number from 1 to 10:");
      getInput();
      if (evaluateGuess()) {
         win = true;
         displayYouWon();
      } else {
         displayClue();
      }
    } while (win == false);
}
}
```

Java Virtual Machine (VM)



Java Compared to C++

Feature	Description
Syntax	Similar to the syntax for C++.
Platforms	After a Java program has been compiled, it can be run on any platform that has a Java interpreter.
Speed	C++ runs faster than Java, partly because it is compiled for a specific platform.
Memory	Most memory operations are handled automatically by Java.

Java Compared to C#

Feature	Description
Syntax	Java syntax is similar to C# syntax.
Platforms	Like compiled Java code, compiled C# code (MSIL) can be run on any system that has the appropriate interpreter.
Speed	C# GUI runs slighter faster than Java GUI.
Memory	Both C# and Java handle most memory operations automatically.

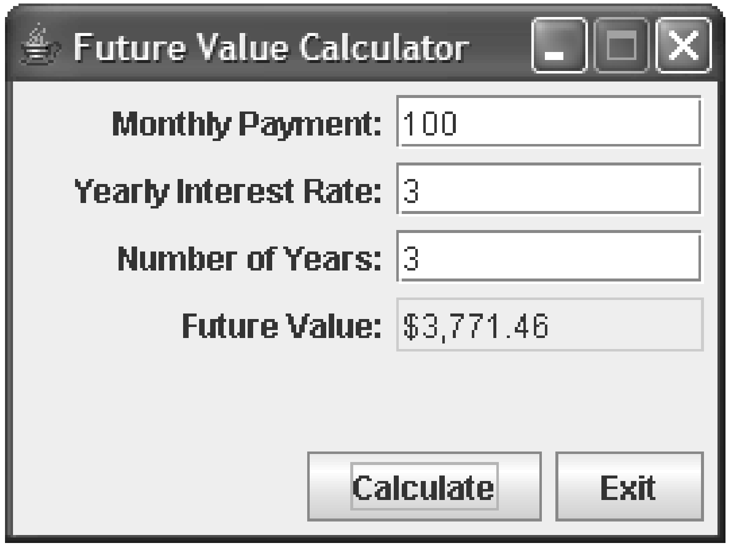
Applications, Applets, & Servlets

An *application* is a program that runs in a window.

An *applet* is a special type of program that runs within a web browser after it has been retrieved from the Internet or an intranet.

A *servlet* is a special type of program that does server-side processing.

Java Application



Future Value Calculator

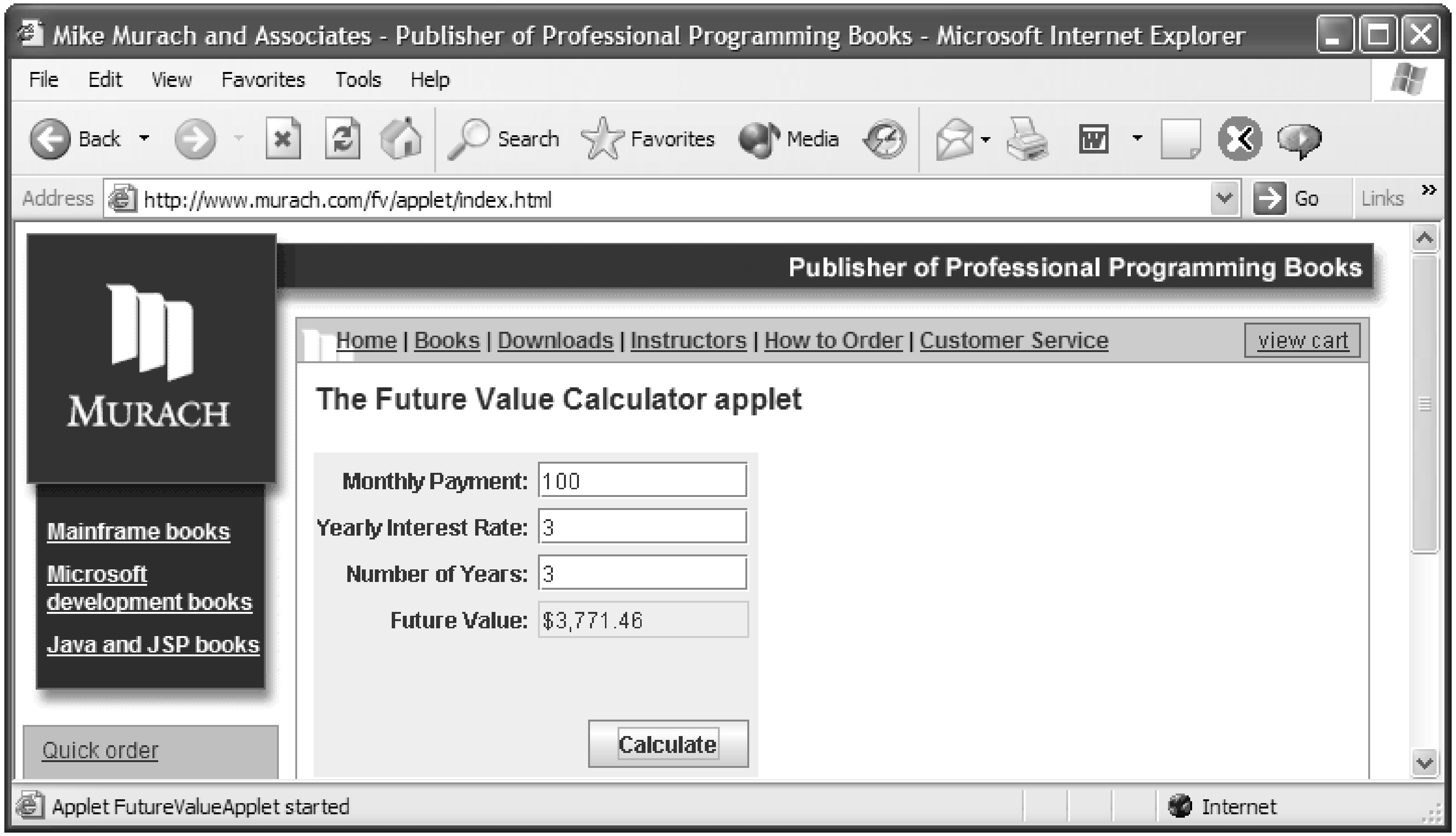
Monthly Payment: 100

Yearly Interest Rate: 3

Number of Years: 3

Future Value: $3,771.46

Java Applet

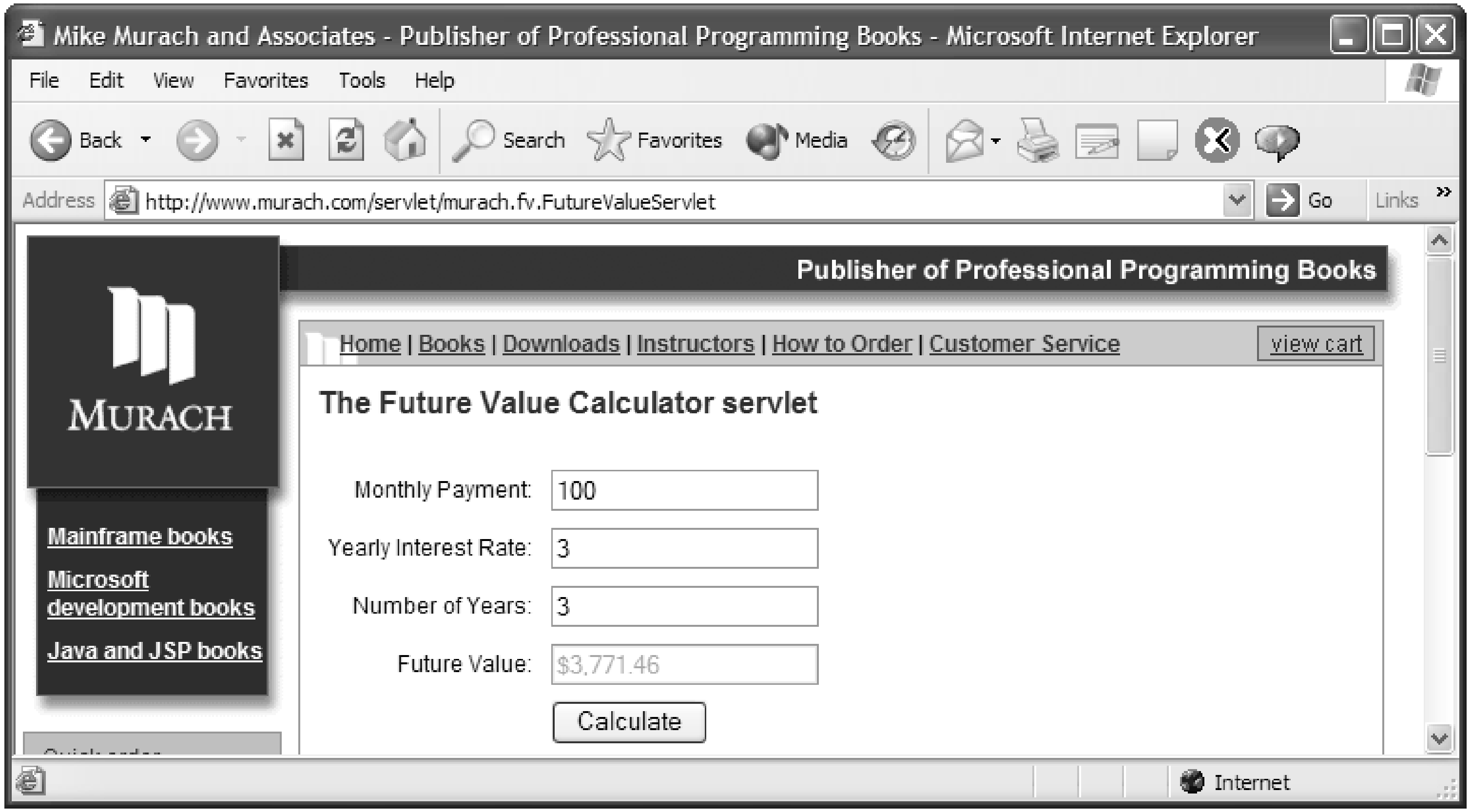

Mike Murach and Associates - Publisher of Professional Programming Books - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://www.murach.com/fv/applet/index.html
Go
Links
Publisher of Professional Programming Books
MURACH
Home | Books | Downloads | Instructors | How to Order | Customer Service
view cart
Mainframe books
Microsoft development books
Java and JSP books
Quick order
Calculate
Internet

The Future Value Calculator applet

Monthly Payment:	100
Yearly Interest Rate:	3
Number of Years:	3
Future Value:	$3,771.46

Applet FutureValueApplet started

Java Servlet

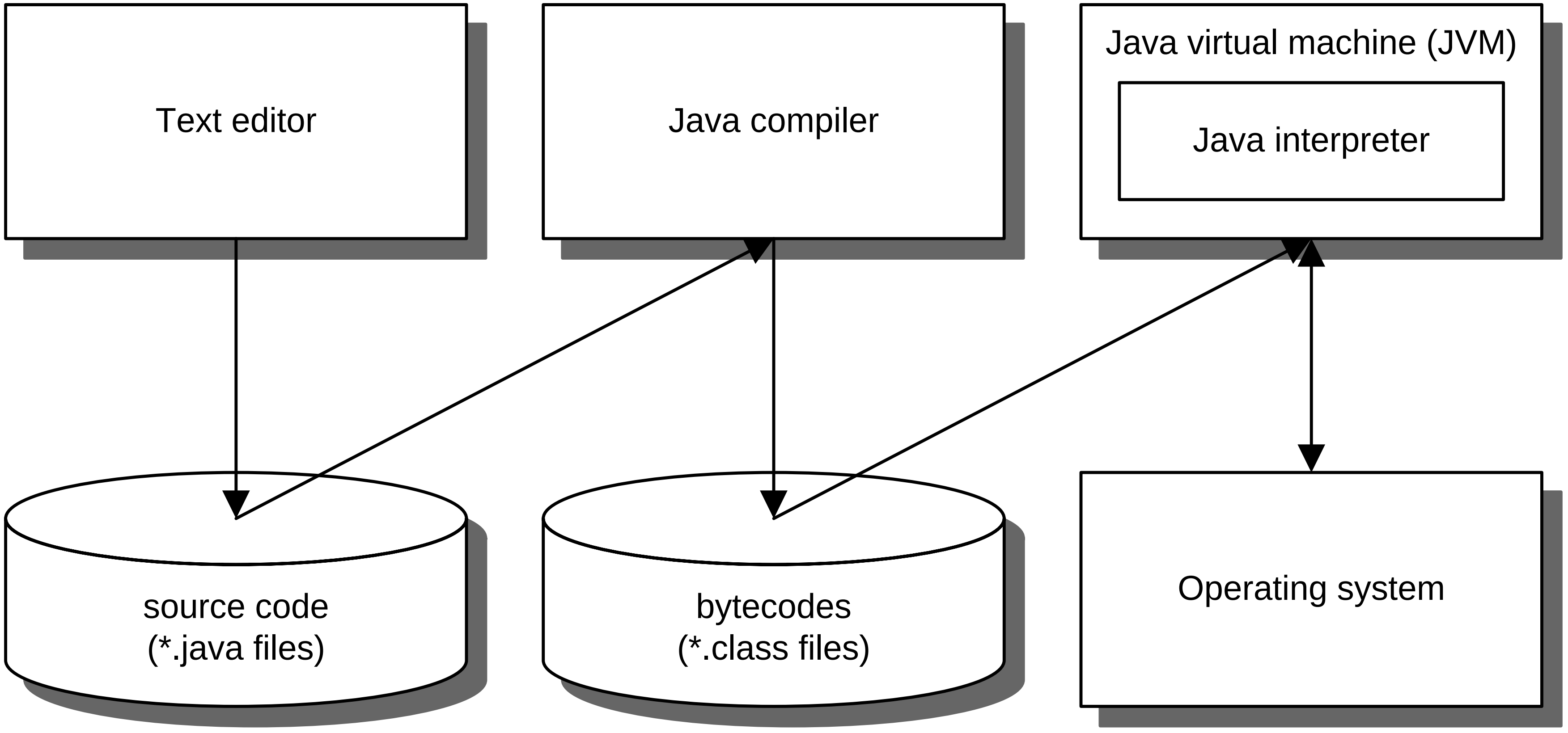

Mike Murach and Associates - Publisher of Professional Programming Books - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go
Links
Publisher of Professional Programming Books
MURACH
Home | Books | Downloads | Instructors | How to Order | Customer Service
view cart
Mainframe books
Microsoft development books
Java and JSP books
Calculate
Internet

Address: http://www.murach.com/servlet/murach.fv.FutureValueServlet

The Future Value Calculator servlet

Monthly Payment:	100
Yearly Interest Rate:	3
Number of Years:	3
Future Value:	$3,771.46

How Java Interprets & Executes Code



How Java Interprets & Executes Code

Classes, bytecodes, and JVMs

- To develop a Java application, you develop one or more *classes*.
- You can use any text editor to create, edit, and save the source code for a Java class. Source code files have the *java* extension.
- The *Java compiler* translates Java source code into a *platform-independent* format known as Java *bytecodes*. Files that contain Java bytecodes have the *class* extension.
- The *Java interpreter* executes Java bytecodes. Java interpreters exist for all major operating systems.
- A Java interpreter is an implementation of a *Java virtual machine* (*JVM*).

Java SE (Standard Edition)

Java SE is distributed for free by the Oracle Corporation

Two distributions

- JRE (Java Runtime Edition) for running applications and applets
- JDK (Java Development Kit) for developing java programs
  - includes the JRE

Available from http://www.oracle.com/technetwork/java/javase/overview/index.html

The current version is 1.9.0

As a programmer you download the JDK

NetBeans

Integrated Development Environment

Bundles a wide range of tools

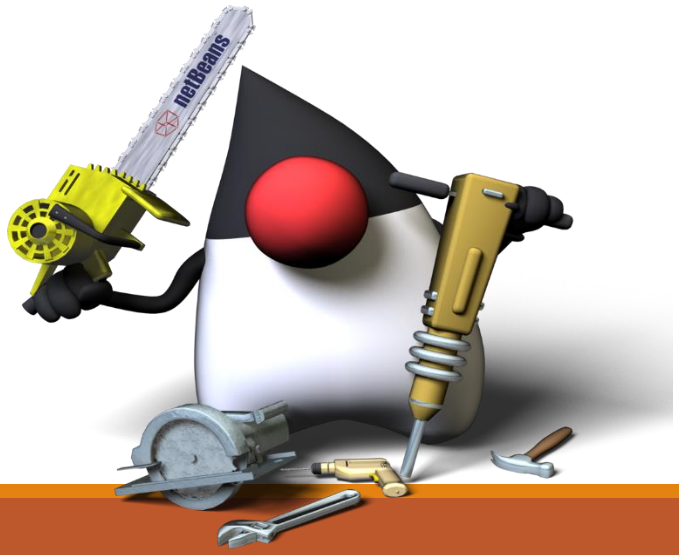
Used to build all types of Java applications

Syntax checking and code hints as you type

Other IDEs include Eclipse, IntelliJ, JDeveloper, and JBuilder

Available from http://netbeans.org

Current version is 8



"Example isn’t another way to teach, it is the only way to teach“

ALBERT EINSTEIN