Java Fundamentals

Training @ Basis

Course objectives

- Introduction to Java programming language
- Object Oriented Programming in Java
- Java API – Strings, I/O, Properties
- Relational database concepts and JDBC
- Internationalization
- Introduction to web application programming using Java EE

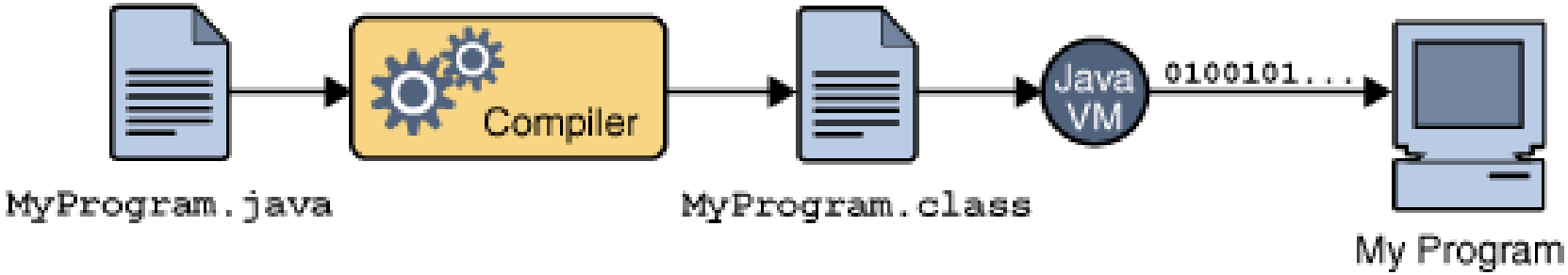
What is a computer?

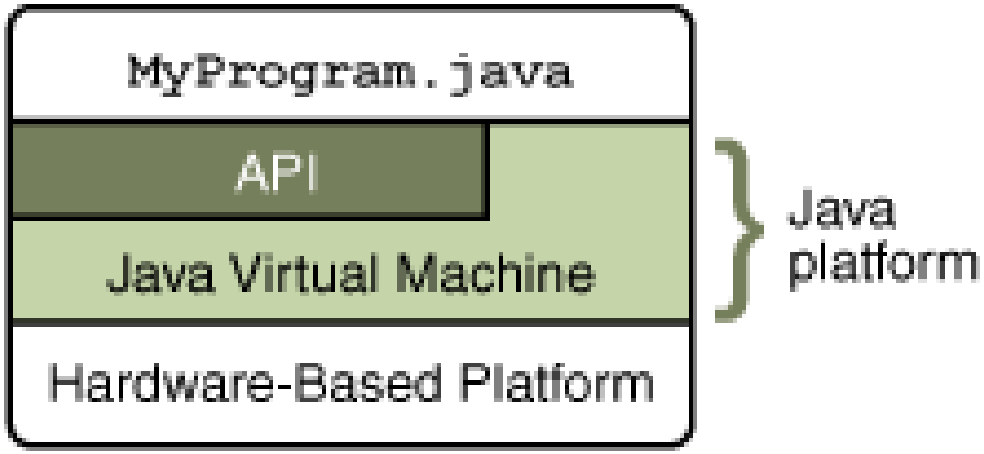
Input → Process → Output

A Java Premier

- What is Java? - A programming language & a platform
- Why Java? - Benefits to the industry, businesses and individuals and software developers.
- Installing JDK, JRE
- Environment variables: JAVA_HOME, PATH

The Java programming language



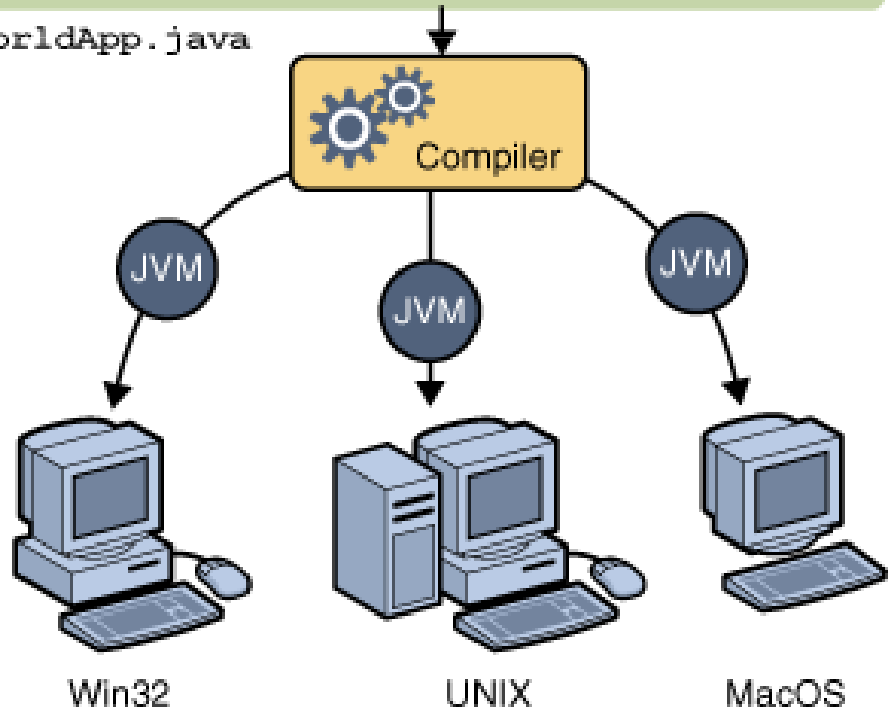


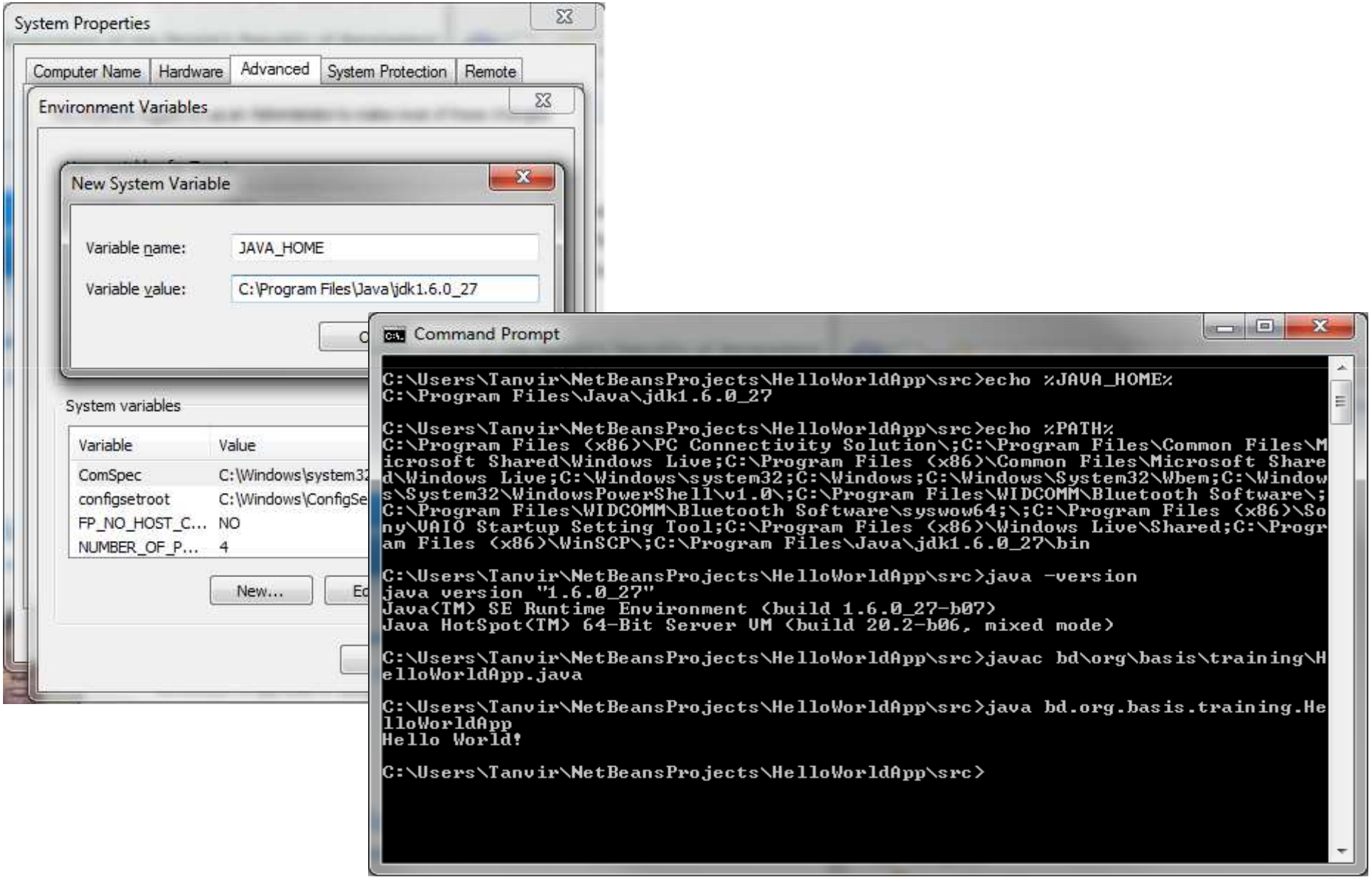
Say “Hello World!” in Java

Java Program

```
class HelloWorldApp {
     public static void main(String[] args) {
          System.out.println("Hello World!");
     }
}
```

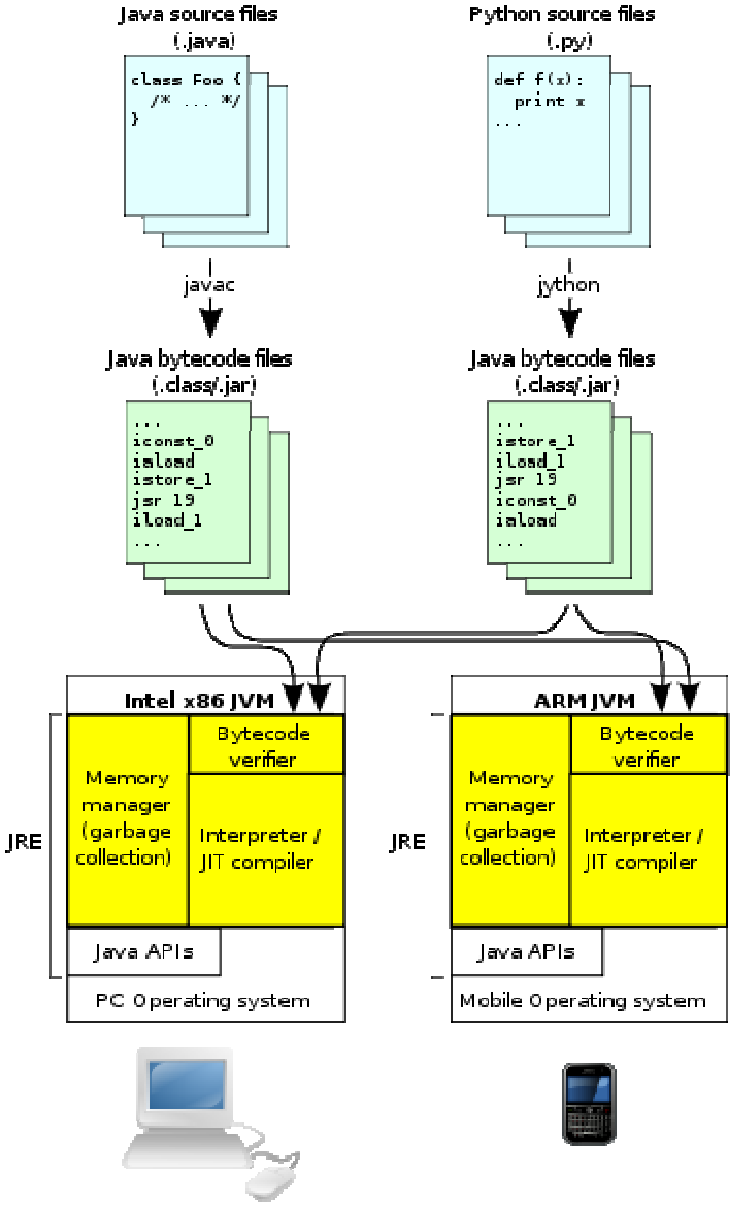
HelloWorldApp.java




System Properties
Computer Name
Hardware
Advanced
System Protection
Remote
Environment Variables
New System Variable
Variable name:
JAVA_HOME
Variable value:
C:\Program Files\Java\jdk1.6.0_27
System variables
Variable
Value
ComSpec
C:\Windows\system32
configsetroot
C:\Windows\ConfigSe
FP_NO_HOST_C...
NO
NUMBER_OF_P...
4
New...
Command Prompt
C:\Users\Tanvir\NetBeansProjects\HelloWorldApp\src>echo %JAVA_HOME%
C:\Program Files\Java\jdk1.6.0_27
C:\Users\Tanvir\NetBeansProjects\HelloWorldApp\src>echo %PATH%
C:\Program Files (x86)\PC Connectivity Solution\;C:\Program Files\Common Files\Microsoft Shared\Windows Live;C:\Program Files (x86)\Common Files\Microsoft Shared\Windows Live;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Program Files\WIDCOMM\Bluetooth Software\;C:\Program Files\WIDCOMM\Bluetooth Software\syswow64;\;C:\Program Files (x86)\Sony\VAIO Startup Setting Tool;C:\Program Files (x86)\Windows Live\Shared;C:\Program Files (x86)\WinSCP\;C:\Program Files\Java\jdk1.6.0_27\bin
C:\Users\Tanvir\NetBeansProjects\HelloWorldApp\src>java -version
java version "1.6.0_27"
Java(TM) SE Runtime Environment (build 1.6.0_27-b07)
Java HotSpot(TM) 64-Bit Server VM (build 20.2-b06, mixed mode)
C:\Users\Tanvir\NetBeansProjects\HelloWorldApp\src>javac bd\org\basis\training\HelloWorldApp.java
C:\Users\Tanvir\NetBeansProjects\HelloWorldApp\src>java bd.org.basis.training.HelloWorldApp
Hello World!
C:\Users\Tanvir\NetBeansProjects\HelloWorldApp\src>

"Java Virtual Machine“ (JVM)

- An abstraction layer on top an operating system platform.
- Cross-platform/platform independence.
- Executes bytecode from .class files
- "write once, run anywhere"


Java source files
(.java)
class Foo {
/* ... */
}
Python source files
(.py)
def f(x):
print x
...
javac
jython
Java bytecode files
(.class/.jar)
...
iconst_0
imload
istore_1
jsr 19
iload_1
...
Java bytecode files
(.class/.jar)
...
istore_1
iload_1
jsr 19
iconst_0
imload
...
Intel x86 JVM
Bytecode verifier
Memory manager (garbage collection)
Interpreter / JIT compiler
Java APIs
PC Operating system
JRE
ARM JVM
Bytecode verifier
Memory manager (garbage collection)
Interpreter / JIT compiler
Java APIs
Mobile Operating system
JRE

Integrated Development Environment (IDE)

- All-in-one package to development, test, debug, profile and deploy an application
- Popular IDEs
  - NetBeans http://netbeans.org/
  - Eclipse http://eclipse.org/
  - IntelliJ IDEA http://www.jetbrains.com/idea/

Eclipse

Java - HelloWorldApp.java - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explo... Hierarchy

HelloWorldApp
src
bd.org.basis.pooj
HelloWorldApp.java
JRE System Library [jre]

*HelloWorldApp.java

```
package bd.org.basis.pooj;

public class HelloWorldApp {

    /**
     * @param args
     */
    public static void main(Strin
        System.out.println("Hello
```

Outline

bd.org.basis.pooj
HelloWorldApp
main(String[])

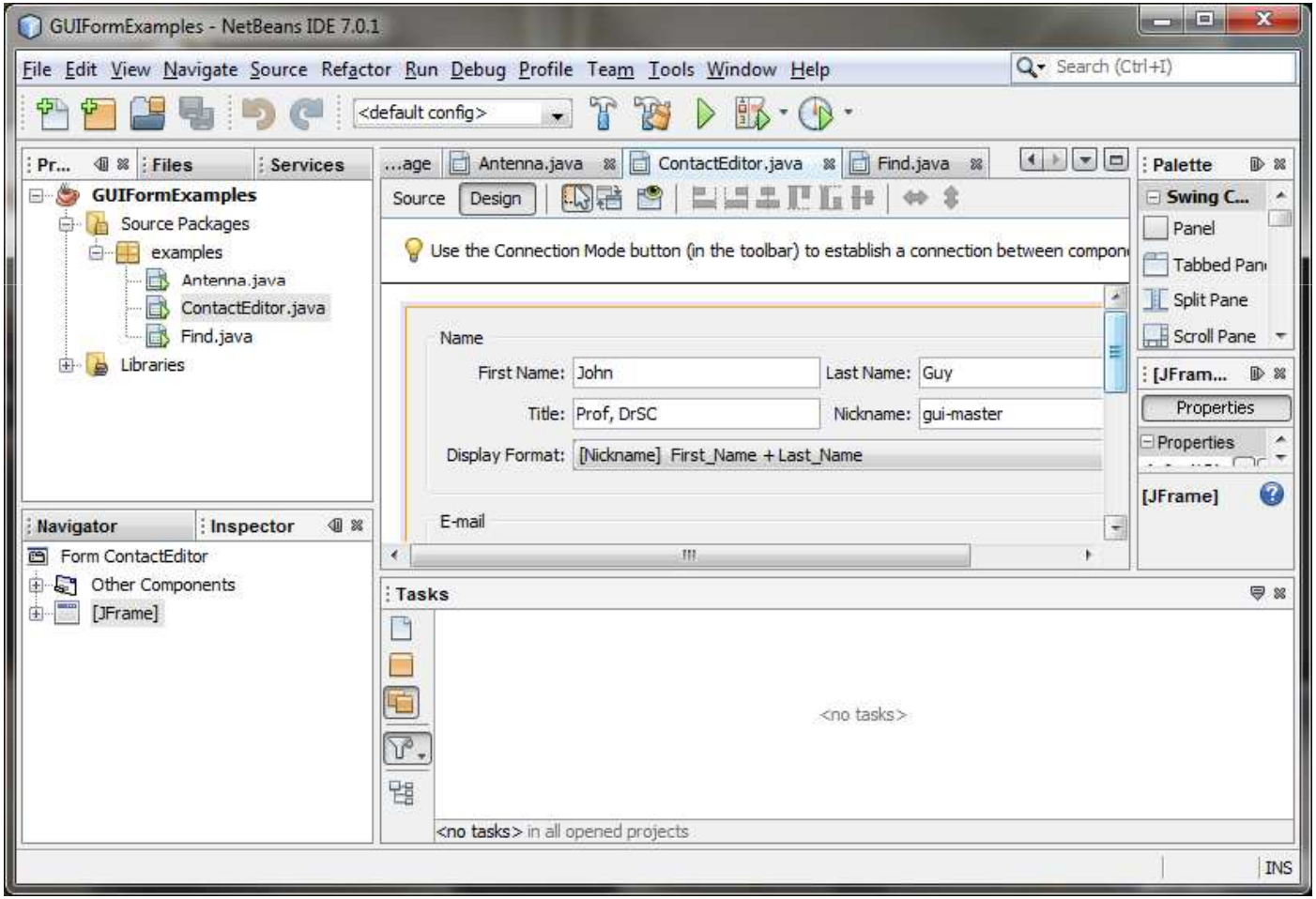
Problems Javadoc Declaration Console

<terminated> HelloWorldApp [Java Application] C:\F

```
Hello World!
```

Writable Smart Insert 13 : 1

NetBeans


GUIFormExamples - NetBeans IDE 7.0.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Pr... Files Services
GUIFormExamples
Source Packages
examples
Antenna.java
ContactEditor.java
Find.java
Libraries
Navigator Inspector
Form ContactEditor
Other Components
[JFrame]
...age Antenna.java ContactEditor.java Find.java
Source Design
Use the Connection Mode button (in the toolbar) to establish a connection between compon
Name
First Name: John
Last Name: Guy
Title: Prof, DrSC
Nickname: gui-master
Display Format: [Nickname] First_Name + Last_Name
E-mail
Palette
Swing C...
Panel
Tabbed Pan
Split Pane
Scroll Pane
[JFram...
Properties
Properties
[JFrame]
Tasks
<no tasks>
<no tasks> in all opened projects
INS

Standards

- Code Conventions for the Java Programming Language
  - http://www.oracle.com/technetwork/java/codeconv-138413.html
  - http://www.oracle.com/technetwork/java/codeconventions-137946.html#182
- Java Community Process (JCP)
  - http://jcp.org

The Java Language

- Packages
- Classes
- Variables
- Operators
- Methods
- Static
- Final
- Expression
- Statements
- Blocks
- Control statements
- Interfaces
- Exceptions
- Comments

Data types

- int
- byte
- short
- long
- float
- double
- boolean
- char

Keyword	Description	Size/Format
byte	Byte-length integer	8-bit two's complement
short	Short integer	16-bit two's complement
int	Integer	32-bit two's complement
long	Long integer	64-bit two's complement
float	Single-precision floating point	32-bit IEEE
double	Double-precision floating point	64-bit IEEE
char	A single character	16-bit Unicode character
boolean	A boolean value (true or false)	true or false

Literals

Type	Sample values
Integer Literals	10, 659L, 0x4a, 057L
Character Literals	'a', '#', '3', '\\', '\n', 'u0041'
Boolean Literals	true, false
Floating-point literals	4.13179, -0.000001, 7D, 0.01f
String Literals	"How are you?" "" // the empty string "\"" // a string containing "
Null Literals	null

Operators

Category	Operators
Simple Assignment Operator	=
Arithmetic Operators	**+** Additive operator (also used for String concatenation) - Subtraction operator * Multiplication operator / Division operator % Remainder operator
Conditional Operators	&& Conditional-AND || Conditional-OR ?: Ternary (shorthand for if-then-else statement)
Type Comparison Operator	instanceof Compares an object to a specified type

Operators (cont…)

Category	Operators
Unary Operators	+ Unary plus; indicates positive value (default) - Unary minus; negates an expression ++ Increment; increments a value by 1 -- Decrement; decrements a value by 1 ! Logical compliment; inverts the value of a boolean
Equality and Relational Operators	== Equal to != Not equal to > Greater than >= Greater than or equal to < Less than <= Less than or equal to

Operators (cont…)

Category	Operators
Bitwise and Bit Shift Operators	~ Unary bitwise complement << Signed left shift >> Signed right sift >>> Unsigned right shift & Bitwise AND ^ Bitwise exclusive OR | Bitwise inclusive OR

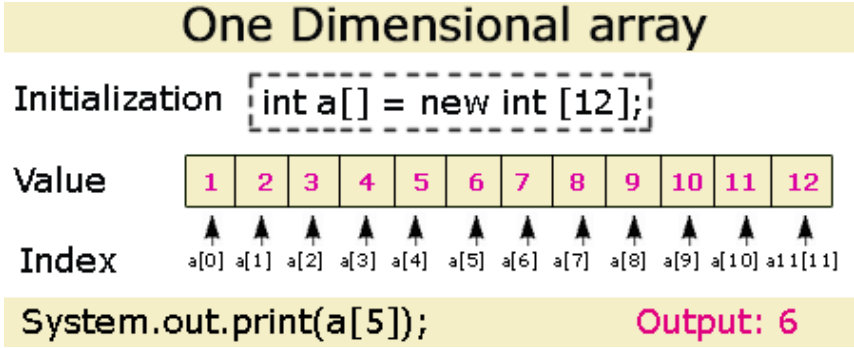
Operators Precedence

Order	Operators	Precedence
1	postfix	expr++,, expr--
2	unary	++expr, --expr, +expr, -expr ~ !
3	multiplicative	* / %
4	additive	+ -
5	shift	<< >> >>>
6	relational	< > , <= , >= instanceof
7	equality	== , !=
8	bitwise AND	&

Order	Operators	Precedence
9	bitwise exclusive OR	^
10	bitwise inclusive OR	|
11	logical AND	&&
12	logical OR	||
13	ternary	? :
14	assignment	= , +=, -=, *=, /=, %=, &=, ^=, |=, <<=, >>=, >>, >=

Arrays

- Store the values of the same type
  in contiguous memory allocations.
- Always a fixed length abstracted data structure which can not be altered when required.



Using arrays

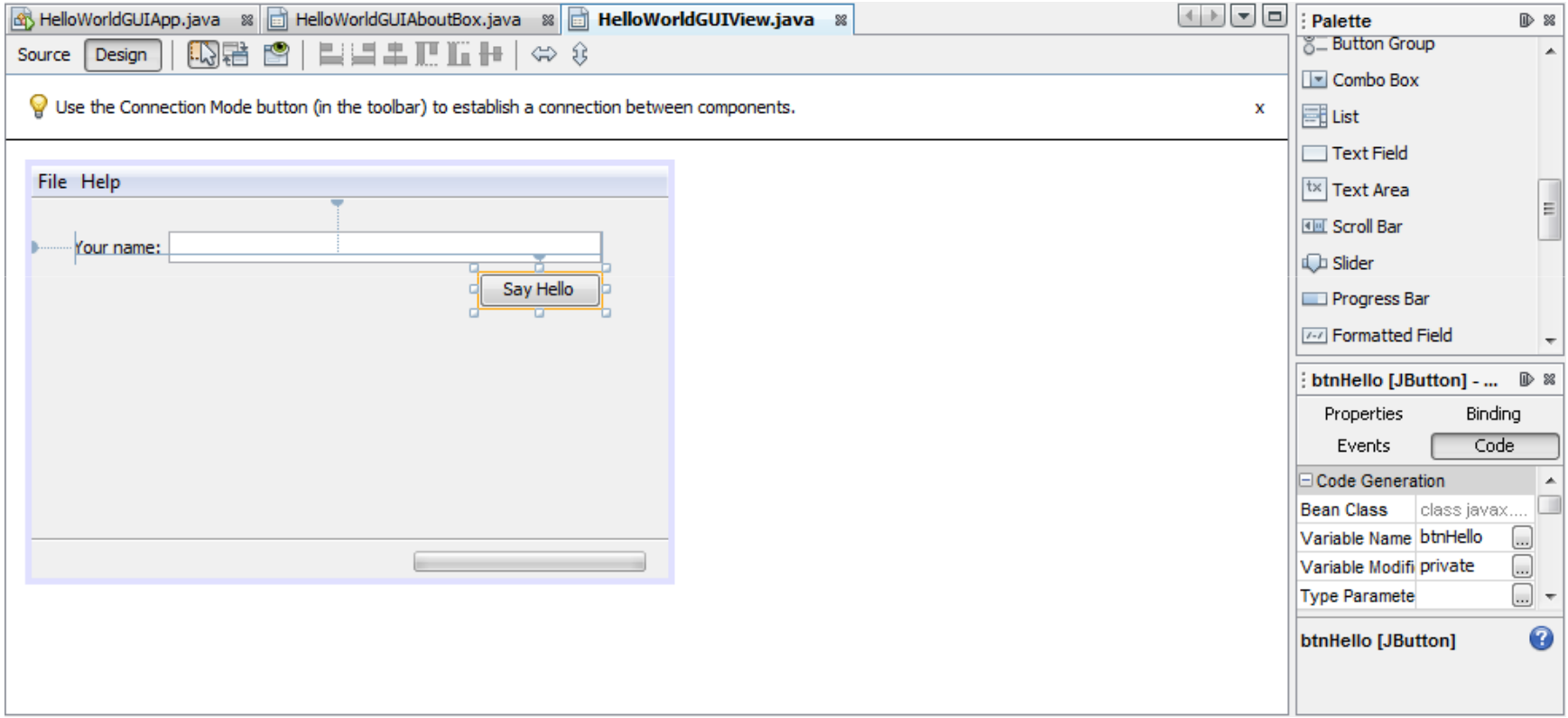
...ava | HelloWorldApp.java | ArraySum.java | PrintMonths.java

```java
5   package bd.org.basis.training;
6
7   /**
8    *
9    * @author Tanvir
10   */
11  public class ArraySum {
12
13      public static void main(String[] args) {
14          int[] x = new int[101];
15          for (int i = 0; i < x.length; i++) {
16              x[i] = i;
17          }
18          int sum = 0;
19          for (int i = 0; i < x.length; i++) {
20              sum += x[i];
21          }
22          System.out.println(sum);
23
24      }
25  }
```

...ava | HelloWorldApp.java | ArraySum.java | PrintMonths.java

```java
5   package bd.org.basis.training;
6
7   /**
8    *
9    * @author Tanvir
10   */
11  public class PrintMonths {
12
13      public static void main(String[] args) {
14          String months[] = {"Jan", "Feb", "Mar", "Apr", "May", "Jun"
15              "July", "Aug", "Sep", "Oct", "Nov", "Dec"};
16          //use the length attribute to get the number
17          //of elements in an array
18          for (int i = 0; i < months.length; i++) {
19              System.out.println("month: " + months[i]);
20          }
21      }
22  }
23
```

Hello World in Windows



Hello World in Windows

HelloWorldGUIApp.java | HelloWorldGUIAboutBox.java | **HelloWorldGUIView.java**

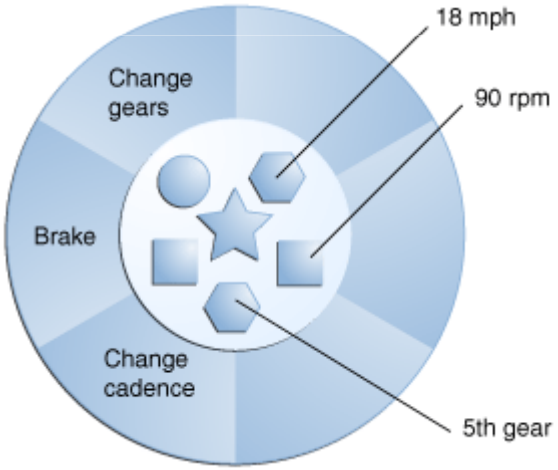
Source | Design

```
225 private void btnHelloActionPerformed(java.awt.event.ActionEvent evt) {
226     String message = null;
227     int messageType = JOptionPane.QUESTION_MESSAGE;
228     if (txtName.getText().length() == 0) {
229         message = "Hello you! What's your name?";
230     } else {
231         message = "Hello " + txtName.getText() + "!";
232         messageType = JOptionPane.INFORMATION_MESSAGE;
233     }
234     JOptionPane.showMessageDialog(this.getFrame(), message, "Hello GUI", messageType);
235 }
236
237     // Variables declaration - do not modify
238     private javax.swing.JButton btnHello;
239     private javax.swing.JLabel jLabel1;
240     private javax.swing.JPanel mainPanel;
241     private javax.swing.JMenuBar menuBar;
242     private javax.swing.JProgressBar progressBar;
243     private javax.swing.JLabel statusAnimationLabel;
244     private javax.swing.JLabel statusMessageLabel;
245     private javax.swing.JPanel statusPanel;
246     private javax.swing.JTextField txtName;
247     // End of variables declaration
```

Exercise: Develop a standard calculator with graphical user interface and memory functions i.e. a Windows Calculator clone.

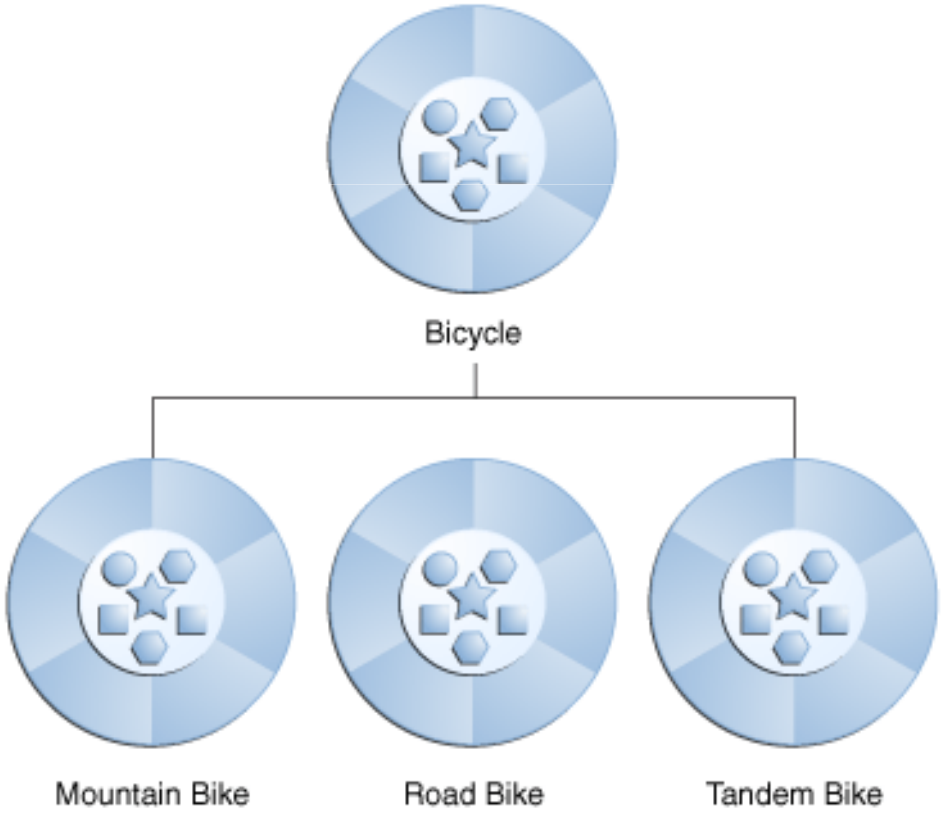
Object Oriented Programming

- An object is a software bundle of related
  - State
  - Behaviour
- OOP
  - Data encapsulation
  - Inheritance
  - Polymorphism



Object Oriented Programming …

- Class
- Inheritance
- Interface
- Package



Java Classes

- Member variables
- Methods
- Constructors
- Instances
- “this” keyword
- Access control
- Abstract classes

Modifier	Class	Package	Subclass	World
Public	Y	Y	Y	Y
Protected	Y	Y	Y	N
No modifier	Y	Y	N	N
Private	Y	N	N	N

Exercise: An Employee Database

- Types of employees
  - Permanent - Salary = Basic + House Rent
  - Temporary
    - Casual – Salary = Daily rate * No. of days worked
    - Consultant – Salary = Monthly rate – 10% AIT
- Name, Phone, Department

Java APIs

- Application Programming Interface - A source code based specification intended to be used as an interface by software components to communicate with each other. An API may include specifications for routines, data structures, object classes, and variables.
- http://java.sun.com/reference/api/

Java APIs (cont…)

- A collection of library routines that performs basic programming tasks such as looping, displaying GUI form etc.
- API classes and interfaces are packaged in packages.
- All these classes are written in Java and runs on the JVM.
- Java classes are platform independent.

Java API Specifications

Standard Edition	Enterprise Edition
Java SE 6 J2SE 1.5.0 J2SE 1.4.2 J2SE 1.3.1	Java EE 5 J2EE 1.4 J2EE 1.3 J2EE 1.2.1
Micro Edition	**JavaFX**
Java ME	JavaFX 1.0
Micro Edition	**JavaFX**
Java ME	JavaFX 1.0
Javacard	**Java Web Services**
Javacard	Java Web Services

Java API Specifications (cont…)

XML
XML

Other Technologies
Java 3D Java Advanced Imaging (JAI) JavaBeans Activation Framework Java Communications JavaMail Java Media Framework (JMF) Java Speech Real-Time Specification for Java (RTSJ) Other Technologies

Java API Documentation

Overview (Java Platform SE

docs.oracle.com/javase/6/docs/api/overview-summary.html

Overview Package Class Use **Tree** **Deprecated** **Index** **Help**

PREV NEXT FRAMES NO FRAMES All Classes

Java™ Platform Standard Ed. 6

Java™ Platform, Standard Edition 6
API Specification

This document is the API specification for version 6 of the Java™ Platform, Standard Edition.

See:

Description

Packages	
java.applet	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
java.awt	Contains all of the classes for creating user interfaces and for painting graphics and images.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between and within applications.
	Drag and Drop is a direct manipulation gesture found in many

Java SE 6 APIs

- http://docs.oracle.com/javase/6/docs/api/overview-summary.html

Packages	Packages
java.lang	java.io
java.math	java.util
java.net	java.sql, javax.sql
java.text	java.awt, javax.swing
javax.xml, org.w3c.dom	javax.crypto
javax.naming	javax.security

Strings

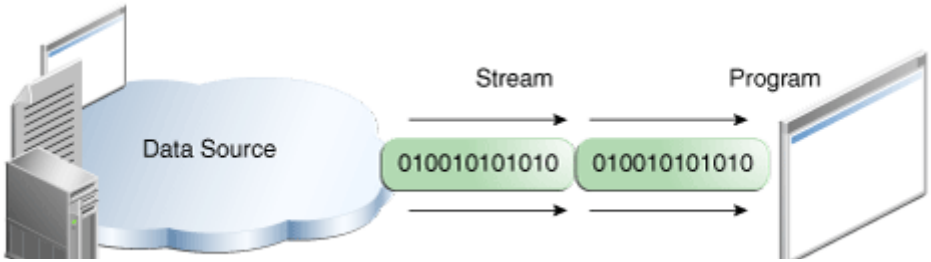
- Class `java.lang.String`
- Construction
  - `char ca[] = {'A', 'E', 'I', 'O', 'U'};`
    `String str = new String(ca);`
  - `String str = new String("AEIOU");`
  - `String str = "AEIOU";`
  - `String str = new String("AEI" + "OU");`
  - `String str = "AEI" + "OU";`

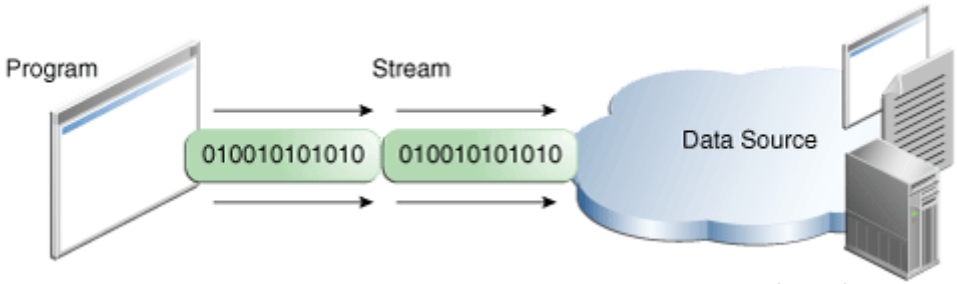
Strings …

- String operations
  - Concatenate, Examine, Compare, Search, Replace, Extract, Convert.
- Two facts:
  - The compiler does not create a duplicate string literal.
  - The modifying methods return a new string.

Using Java I/O API

- Packages – java.io, java.nio
- InputStream, FileInputStream
- OutputStream, FileOutputStream
- Reader, FileReader
- Writer, FileWriter



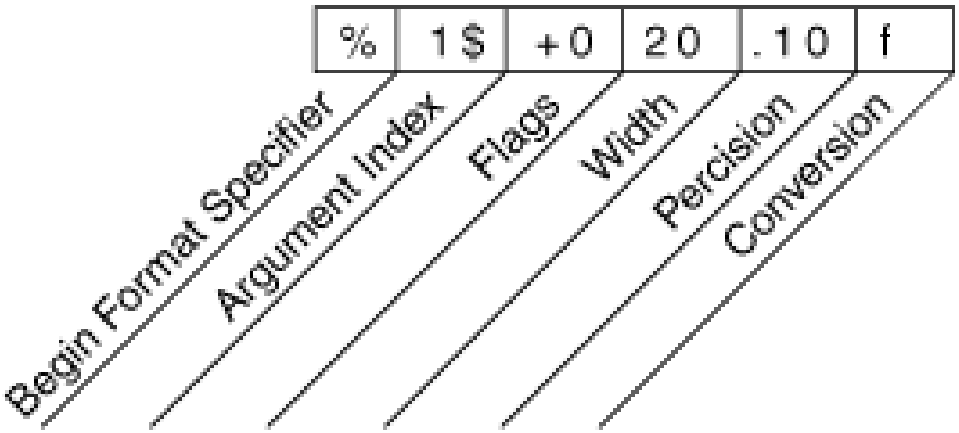


Using Java I/O API (cont…)

- I/O applications
  - Use I/O Streams (CopyBytes)
  - Use Character Streams (CopyCharacters)
  - Use Buffered Streams (CopyCharactersBuffered)
  - Console I/O (PasswordChange)

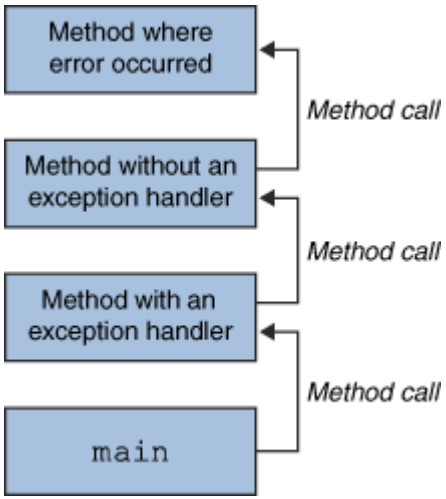
Using Java I/O API (cont…)

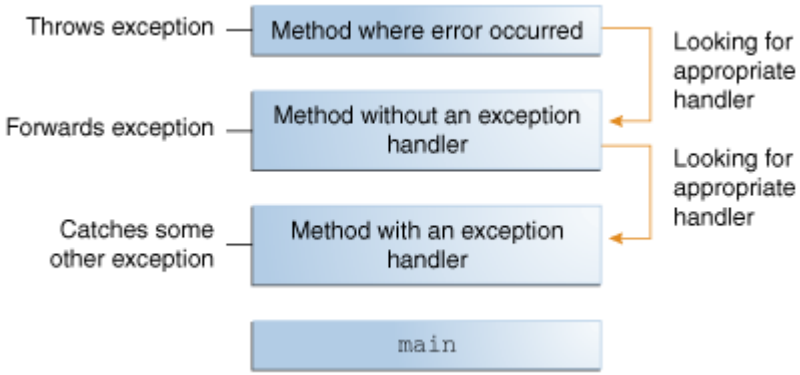
- Scanning and Formatting
  - The Scanner class (ScanSum)
  - The print(), println() and format() methods (SquareRootPi)



Exceptions

- An exception is an event that occurs during the execution of a program that disrupts the normal flow of instructions.





Exceptions (cont…)

```
errorCodeType readFile {
    initialize errorCode = 0;

    open the file;
    if (theFileIsOpen) {
        determine the length of the file;
        if (gotTheFileLength) {
            allocate that much memory;
            if (gotEnoughMemory) {
                read the file into memory;
                if (readFailed) {
                    errorCode = -1;
                }
            } else {
                errorCode = -2;
            }
        } else {
            errorCode = -3;
        }
        close the file;
        if (theFileDidntClose && errorCode == 0) {
            errorCode = -4;
        } else {
            errorCode = errorCode and -4;
        }
    } else {
        errorCode = -5;
    }
    return errorCode;
}
```

```
readFile {
    try {
        open the file;
        determine its size;
        allocate that much memory;
        read the file into memory;
        close the file;
    } catch (fileOpenFailed) {
       doSomething;
    } catch (sizeDeterminationFailed) {
        doSomething;
    } catch (memoryAllocationFailed) {
        doSomething;
    } catch (readFailed) {
        doSomething;
    } catch (fileCloseFailed) {
        doSomething;
    }
}
```

Exceptions (cont…)

- Three kinds of Exceptions
  - Checked Exceptions (**java.lang.Exception)**
  - Error (**java.lang.Error)**
  - Runtime exceptions (**java.lang.RuntimeException)**
- Catching and Handling Exceptions
  - The **`try`**, **`catch`**, and **`finally`** blocks
- Specifying the Exceptions Thrown by a Method

Properties & the Environment

- Properties
  - System.getProperties()
    - Load, Save, Change
  - Exercise – Print and change system properties.
- Environment
  - Map<String, String> env = System.getenv()
  - To maximize portability, prefer system property over environment variable, e.g. user.name.
  - Exercise – Print system environment.

JAR files

Operation	Command
To create a JAR file	jar cf *jar-file input-file(s)*
To create a JAR file with a Manifest	jar cmf *manifest-file jar-file input-file(s)*
To view the contents of a JAR file	jar tf *jar-file*
To extract the contents of a JAR file	jar xf *jar-file*
To extract specific files from a JAR file	jar xf *jar-file archived-file(s)*
To run an application packaged as a JAR file (requires the Main-class manifest header)	java -jar *app.jar*

Relational databases (RDBMS)

- Store, maintain and retrieve data
- Data is presented in tables with rows and columns
- Ensure data integrity
  - Uniqueness
  - Primary key
  - Special 'null' value – 0, blank and null are distinct. 'null' represents absence of data

RDBMS …

- Ensure data integrity …
  - Constraints
  - Foreign keys
- SQL – A language designed to be used with relational databases
  - SELECT
  - INSERT
  - UPDATE
  - DELETE

RDBMS – SQL (DML)

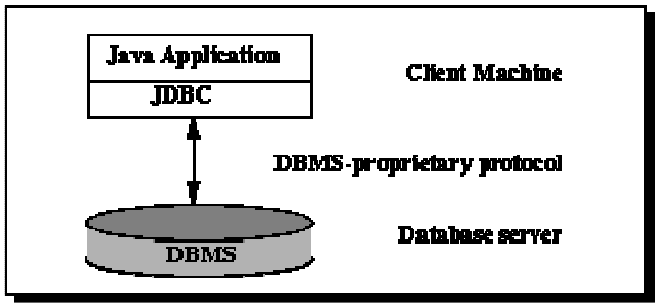
- SELECT *column,…* FROM *table* WHERE *condition*
- INSERT INTO *table (column,…)* VALUES *(value,…)*
- UPDATE *table* SET *column=value,…* WHERE *condition*
- DELETE *table* WHERE *condition*

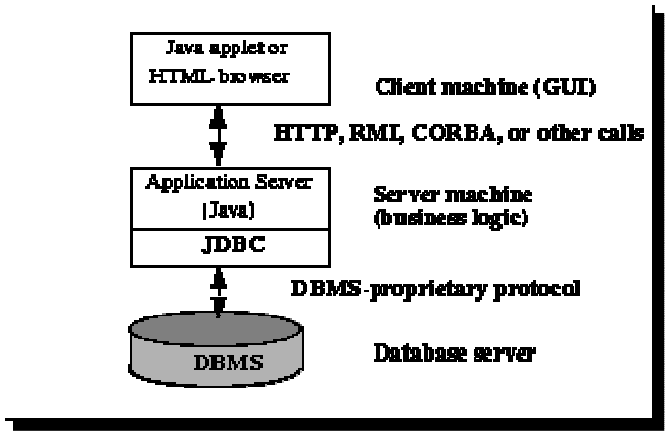
RDBMS – SQL (DDL)

- CREATE TABLE *table (create_definition,…) [table_options]*
- ALTER TABLE *table (alter_specification,…)*
- DROP TABLE *table*
- CREATE INDEX *index [index_type]* ON *table (index_columns,…) [index_options]*
- CREATE VIEW *view [(column_list)]* AS *select_statement*

Database and Java

- JDBC API – Uniform access to RDMBS from Java applications.
  - Multi-tiered architecture
    - Two tier
    - Three tier





JDBC Components

- The JDBC API
  - Connect to a data source
  - Send queries and update statements to the database
  - Retrieve and process the results received from the database in answer to your query
- JDBC Driver Manager
  - DriverManager
  - DataSource

Exercise - HelloJDBC

NetBeans IDE 7.0.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects - J... | Files | Serv...

- MySQL Server at localhost:3306 [root]
- Java DB
- Drivers
- jdbc:metafour://phoenixdb.expotel.local:
- jdbc:mysql://localhost:3306/mystore [roo
  - gov_forms
  - information_schema
  - mysql
  - **mystore**
    - Tables
      - users
        - id
        - userid
        - password
        - firstname
        - lastname
        - email
        - dob
      - Indexes
      - Foreign Keys

HelloJDBC.java

```java
19  public class HelloJDBC {
20
21      private static final Logger logger = Logger.getLogger(HelloJDBC.class.getName());
22
23      /**
24       * @param args the command line arguments
25       */
26      public static void main(String[] args) throws SQLException {
27          logger.info("Hello JDBC!");
28          Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mystore", "root", "root123");
29          Statement stmt = conn.createStatement();
30          ResultSet rs = stmt.executeQuery("SELECT * FROM users");
31          while (rs.next()) {
32              int id = rs.getInt("id");
33              String userid = rs.getString("userid");
34              String password = rs.getString("password");
35              String firstname = rs.getString("firstname");
36              String lastname = rs.getString("lastname");
37              Date dob = rs.getDate("dob");
38              StringBuilder sb = new StringBuilder("");
39              logger.info(sb
40                      .append(id)
```

jdbc:mysql://localhost:3306/test [root on ...

<No View Available>

Tasks | Git Repositories | HTTP Server Monitor | Versioning Output | Search Results

```
run:
02-Feb-2012 11:26:04 hellojdbc.HelloJDBC main
INFO: Hello JDBC!
02-Feb-2012 11:26:05 hellojdbc.HelloJDBC main
INFO: 1,tanvir, secret, Tanvir, Ahmed, null
BUILD SUCCESSFUL (total time: 1 second)
```

Internationalization (i18n)

- ResourceBundle bundle = ResourceBundle.getBundle("training.messages");

```
public static void main(String[] args) {
        ResourceBundle bundle = ResourceBundle.getBundle("training.messages");
        // if specified in the command line load the Locale specific bundle
        if (args.length > 0) {
                bundle = ResourceBundle.getBundle("training.messages", new Locale(args[0]));
        }
        String country_prop = "user.country";
        String language_prop = "user.language";
        Properties props = System.getProperties();
        System.out.println(country_prop + ": " + props.getProperty(country_prop));
        System.out.println(language_prop + ": " + props.getProperty(language_prop));

        System.out.println("Note: You may specify the default language of JVM at startup usin

        System.out.println(bundle.getString("welcome"));
}
```

```
user.country: GB
user.language: bn
Note: You may specify the default language of JVM at startup using system property e.g. -Duser.language=bn
Kemon Aachen?
BUILD SUCCESSFUL (total time: 0 seconds)
```

I18n …

- Use fmt taglib in JSTL
- <fmt:setLocale value="" scope=""/>
- <fmt:setBundle basename="" scope="" var=""/>
- <fmt:message bundle="" key=""/>

Introduction to web applications

- A Java Web application
  - HelloWeb JSP
  - HelloWeb Servlet
- Exercise – Create a Servlet that prints initialisation parameters, form data and HTTP headers.
- Exercise – Create a web form that accepts a name and date of birth of a person and prints the name and age.