

Object-Oriented Concepts and Java

- Object Oriented Concepts
- Object properties
- OO programming overview
- Java and Java Applets
- Distributed objects and Web-Java

Amjad Umar

OBJECT ORIENTED SYSTEMS

- View the world in terms of objects ("things")
- Easier to think of systems as objects (customer, shipping clerk,,)
- main driver: reusability

Example; program with 1 million line of code
effort and cost:

Object-orientation appears as:

- Object-oriented user interfaces (90% of new projects)
- Object-oriented programming (80% of new projects)
- Object-oriented databases (5% or less)
- OO analysis and design (30 to 40%)

Trends:

- Business objects
- Components (VB, JB, EJB)

Copyright (A. Umar) 2002

Object-Oriented Concepts

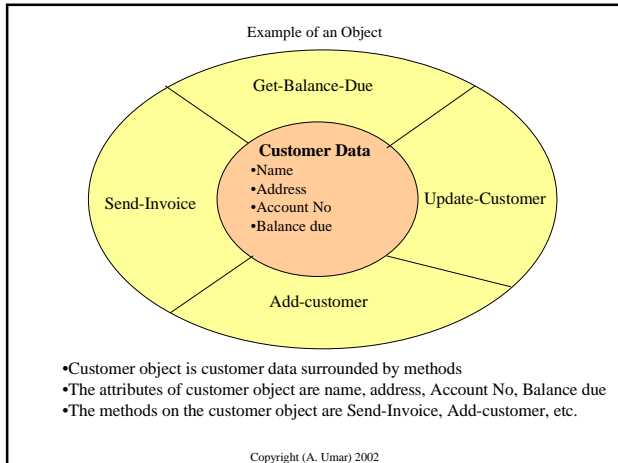
Definition: Objects are data guarded by a protective layer of code
(Methods)

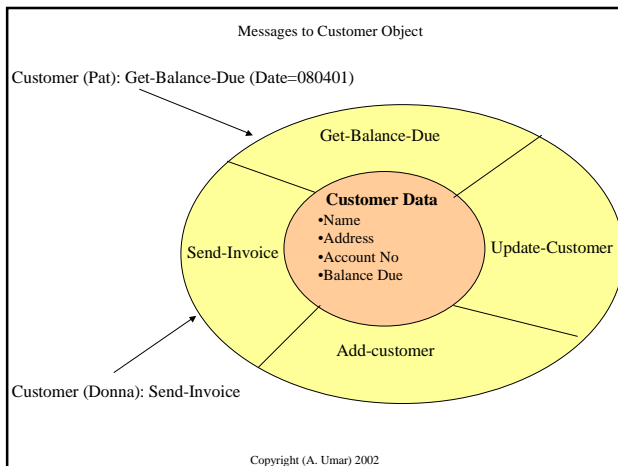
- Key concepts of the object-oriented paradigm:
 - object
 - attributes of objects
 - methods of object show behaviour (what it can provide)
 - class : template of objects
 - inheritance (lower level objects can inherit from higher level)

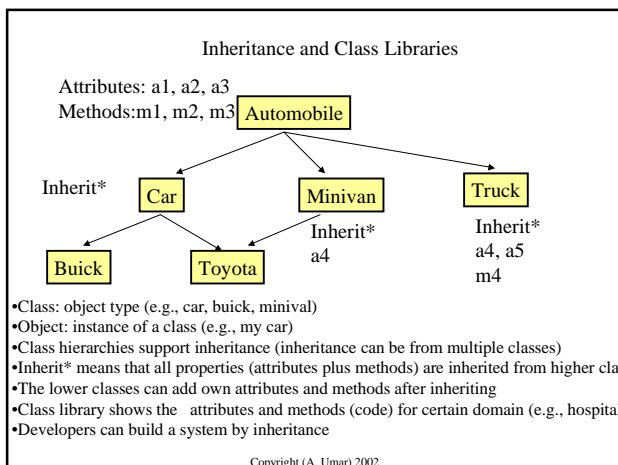
Example: items in a department store :

- TV is an object. The attributes of this object are model, year, cost, etc. The methods, for store are, purchase, repair
- Radio is another object, so is a dishwasher
- many of these objects have similar properties (attributes, methods)
- can define a hierarchy .

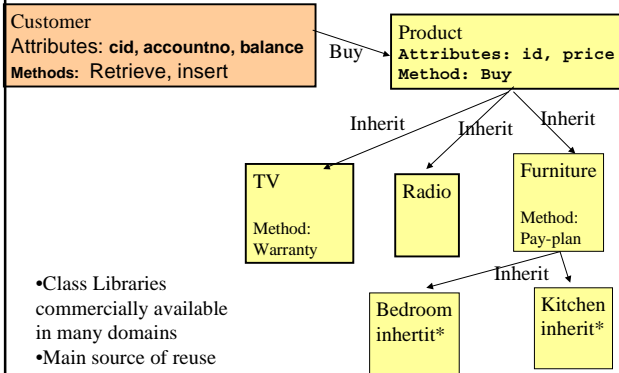
Copyright (A. Umar) 2002







OO Programming Example



Copyright (A. Umar) 2002

Typical OO Program Layout

```
class customer {           // class definition
    int cid, accountno, balance; // private variables
public
    retrieve (int, int);
    insert (int, int, int); }
class product {           // superclass
    id, price, .... // parameters
public
    buy (customer::cid) // method indicating that a customer buys TV }
class Radio: product { // derived class, inherits from product
class TV: product { // derived class; inherits from product
    int warrantor, ..... // parameters
public
    getwarranty () // method
// Methods
    retrieve (int, int); { ..... }
    insert (int, int, int); { ..... }
Main .... The code to initialize objects and perform operations
```

Copyright (A. Umar) 2002

JAVA OVERVIEW

- Designed at Sun initially to develop advanced software for consumer electronics.
- Sun intended to use C++ but found that the problems were best solved by introducing a new language
- Why is Java so hot?
 - Small Java programs, called Java applets, can be embedded in Web pages
 - When users access these pages, these pages along with the Java applets are downloaded to the Web browser.
 - The Java applets run on the Web client side thus making the browser an intelligent component.
 - The Java programs can run on the Web client and can be programmed to do a variety of things.

Copyright (A. Umar) 2002

Java Main Features

- Simplicity: Java was designed as closely to C++ as possible
- Object-Orientation. Essentially those of C++
- Distributed. can open and access objects over the Internet via URL
- Architecture Neutral. Java compiler generates byte codes
- Portable. across platforms
- Java Issues
 - Security issues
 - Performance

Copyright (A. Umar) 2002

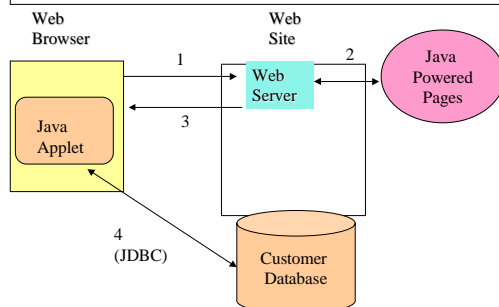
Java Applications and Applets

- Java application
 - a Java code ("Java Class") that has the main () method.
 - Java interpreter looks for main () and executes it
 - standalone application that uses text input and output
- Java applets
 - no main ()
 - run from a Java enabled browser
 - contains methods to initialize itself, draw itself, respond to clicks, etc.
 - is downloaded as part of HTML page (<APPLET Tag)

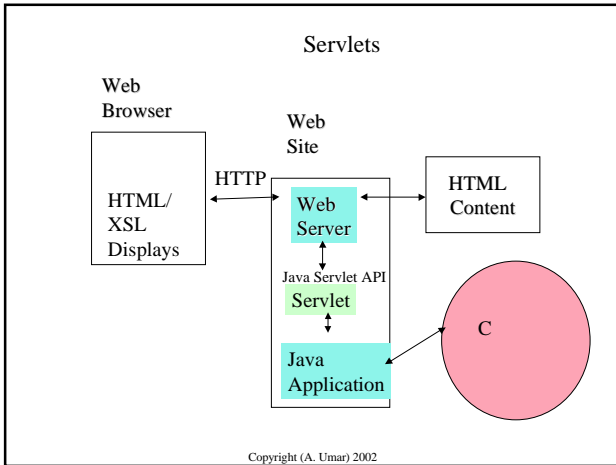
```
<APPLET CODE=applet1.class WIDTH =110 HEIGHT=100>
<PARAM NAME = param1-name VALUE = "param1-name"
</APPLET>
```
 - browser invokes it and runs it

Copyright (A. Umar) 2002

Java Applet



Copyright (A. Umar) 2002



Typical Java Program Layout (Store Example)

```

class customer {           // class definition
int cid, accountno, balance; // private variables
public
  retrieve (int, int);
  insert (int, int, int);
}
class product {           // superclass
  int id, price, .... // parameters
  public buy (cid) // method indicating that a customer buys TV }
class TV Extends product { // derived class; inherits from product
  int warrantor, ..... // parameters
  public getwarranty () // method
class Radio Extends product { // derived class; }
// Methods
int retrieve (int, int); { ..... }
void insert (int, int, int); { ..... }
Main .... The code to initialize objects and perform operations
Sony, RCA = TV //Sony and RCA are instances of TV (objects)
Sony.buy // invoke the method buy on TV Sony
  
```

Copyright (A. Umar) 2002

