Overview: The Mental Landscape

198130 Advanced Computer Programming

Assoc. Prof. Dr. Kanda Runapongsa Saikaew

(krunapon@kku.ac.th)

Department of Computer Engineering

Khon Kaen University

Agenda

- Machine Language
- Asynchronous Events
- The Java Virtual Machine
- Building Blocks of Programs
- Object-oriented Programming
- The Modern User Interface
- The Internet

A computer

- A computer is a complex system consisting of many different components
- The brain of the computer is a single component that does the actual computing
  - The Central Processing Unit, or CPU
- The job of the CPU is to execute programs

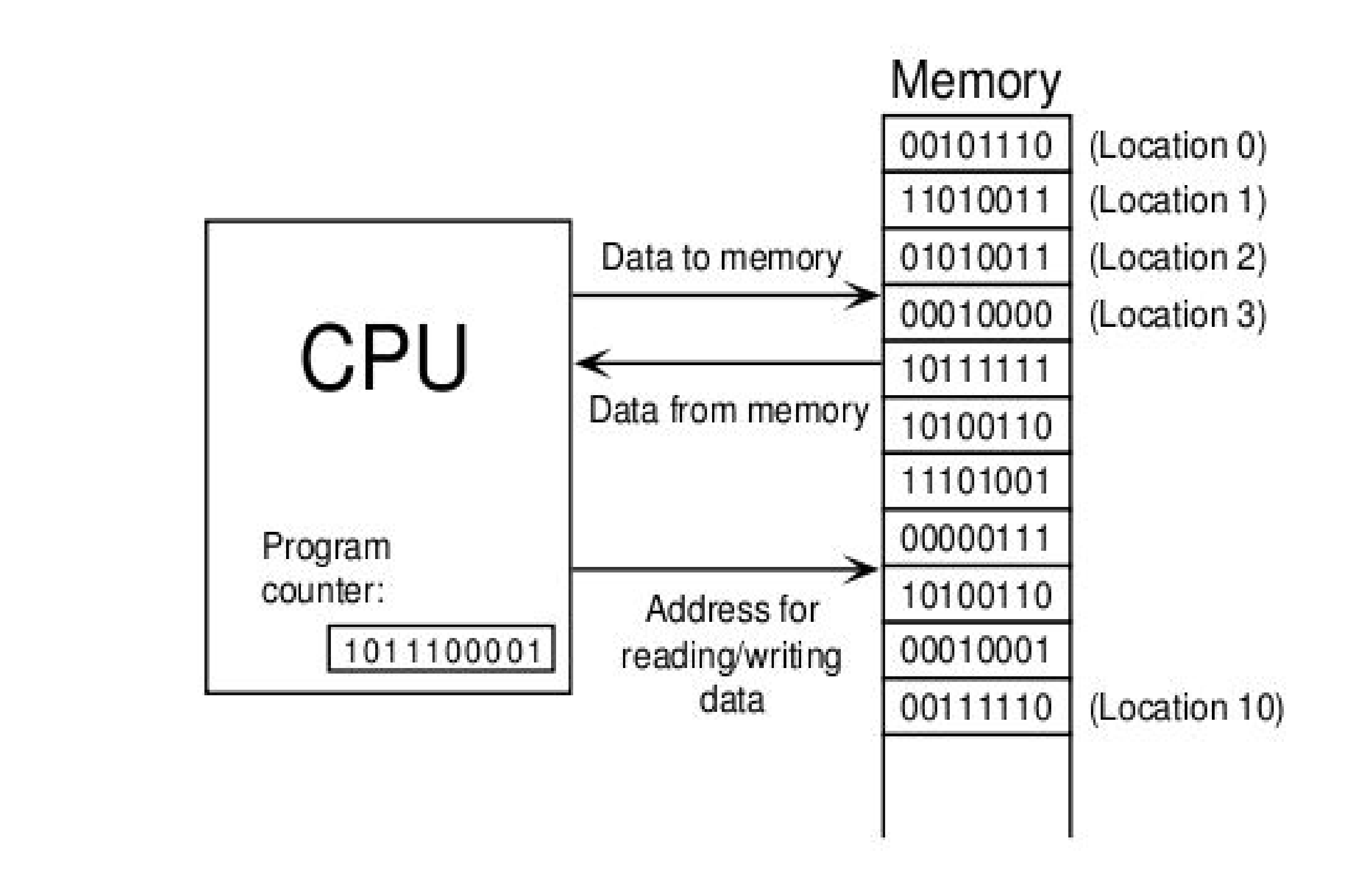
A program

- A program is a list of unambiguous instructions to be followed mechanically by a computer
- A computer is built to carry out instructions that are written in a very simple type of language
  - Machine language
- Each type of computer has its own machine language

How does CPU execute programs?

- CPU executes instructions in memory
- CPU repeatedly reads or fetches an instruction and then executes that instruction
- This process—fetch an instruction, execute it, fetch another instruction, execute it, and so on forever
  - The fetch-and-execute cycle

CPU and Memory



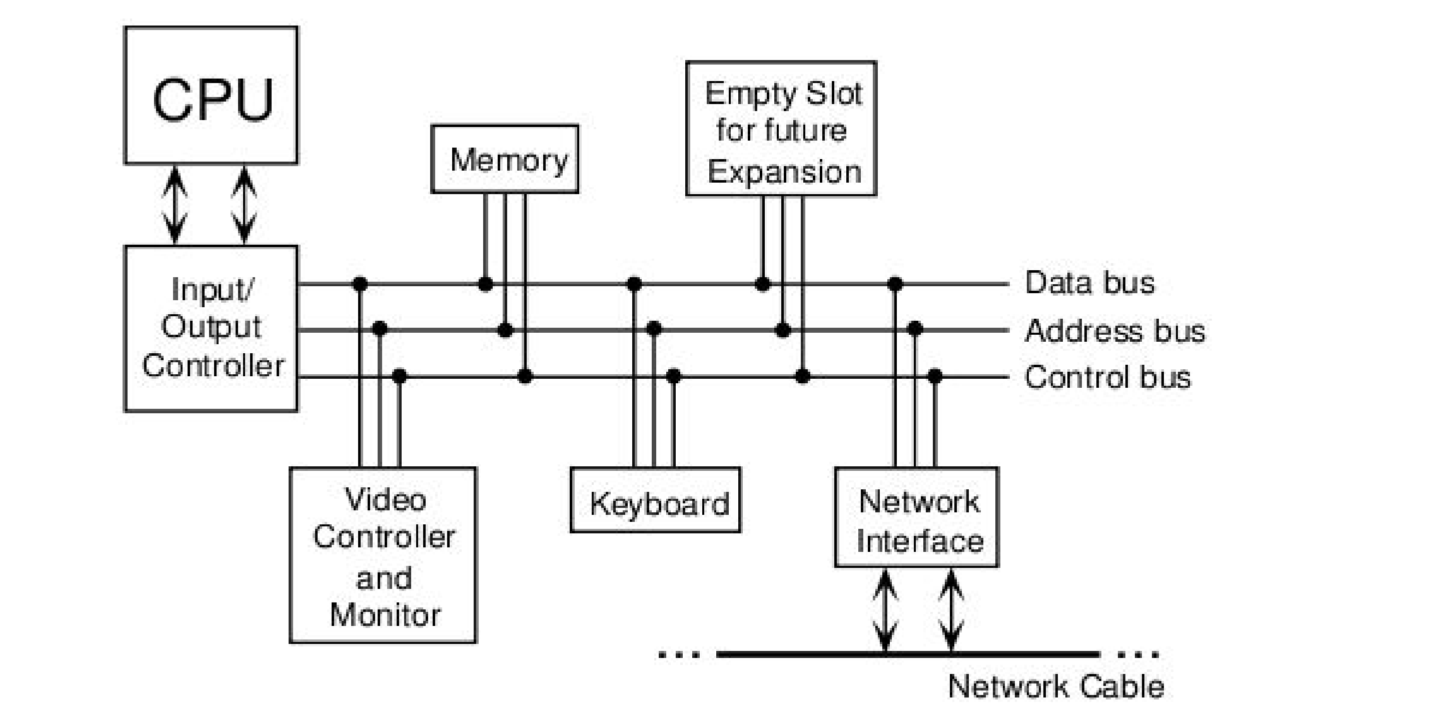
A complete system

- A hard disk for storing programs and data files
- A keyboard and mouse for user input
- A monitor and printer for displaying output

A complete system

- A modem that allows the computer to communicate with other computers over telephone line
- A network interface that allows the computer to communicate with other computers that are connected to it on a network
- A scanner that converts images into coded binary numbers that can be stored and manipulated on the computer

A computer system



Agenda

- Machine Language
- Asynchronous Events
- The Java Virtual Machine
- Building Blocks of Programs
- Object-oriented Programming
- The Modern User Interface
- The Internet

How does the CPU communicate with peripheral devices?

- Polling
  - The CPU keeps checking for incoming data over and over
  - Whenever the CPU finds data, it processes it
- Interrupts
  - An interrupt is a signal sent by another device to the CPU
  - The CPU responds to an interrupt signal by putting aside whatever it is doing in order to respond to the interrupt

Asynchronous Events

- Interrupts allow the CPU to deal with asynchronous events
- Interrupts make it possible for the CPU to deal efficiently with events that happen ”asynchronously” that is at unpredictable times

Multitasking

- All modern computers are multitasking to perform several tasks at once
- Since the CPU is so fast, it can quickly switch its attention from one user to another, devoting a fraction of a second to each user in turn
- The application of multitasking is timesharing
- A modern PC with a just single user also uses multitasking
  – The user types a paper while a clock is continuously displaying the time and a file is being downloaded over the network

Threads

- Each of the individual tasks that the CPU is working on is called a thread
- At any given time, only one thread can actually be executed by a CPU
- The CPU will continue run the same thread until one of several things happen
  - The thread might **yield** voluntarily to give other threads a chance to run

Threads

- The thread might have to wait for some asynchronous event to occur
  - The thread might request some data from the disk drive, or it might wait for the user to press a key
- The thread might use up its allotted slice of time and be suspended to allow other threads to run
  - Preemptive multitasking
- Threads are built into the Java programming language as a fundamental programming concept

Agenda

- Machine Language
- Asynchronous Events
- The Java Virtual Machine
- Building Blocks of Programs
- Object-oriented Programming
- The Modern User Interface
- The Internet

High-level Language

- Machine language consists of simple instructions that can be executed directly by the CPU
- A program written in a high-level language cannot be run directly on any computer
- High-level language programs have to be translated into a machine language
  - This translation can be done by a program called a compiler

Compilers & Interpreters

- There is an alternative to compiling a high-level language program
- Instead of using a compiler, which translates the program all at once
  - You can use an interpreter which translates instruction-by-instruction, as necessary.
- An interpreter is a program that acts much like a CPU, with a kind of fetch-and-execute cycle
- The designers of Java chose to use a combination of compilation and interpretation

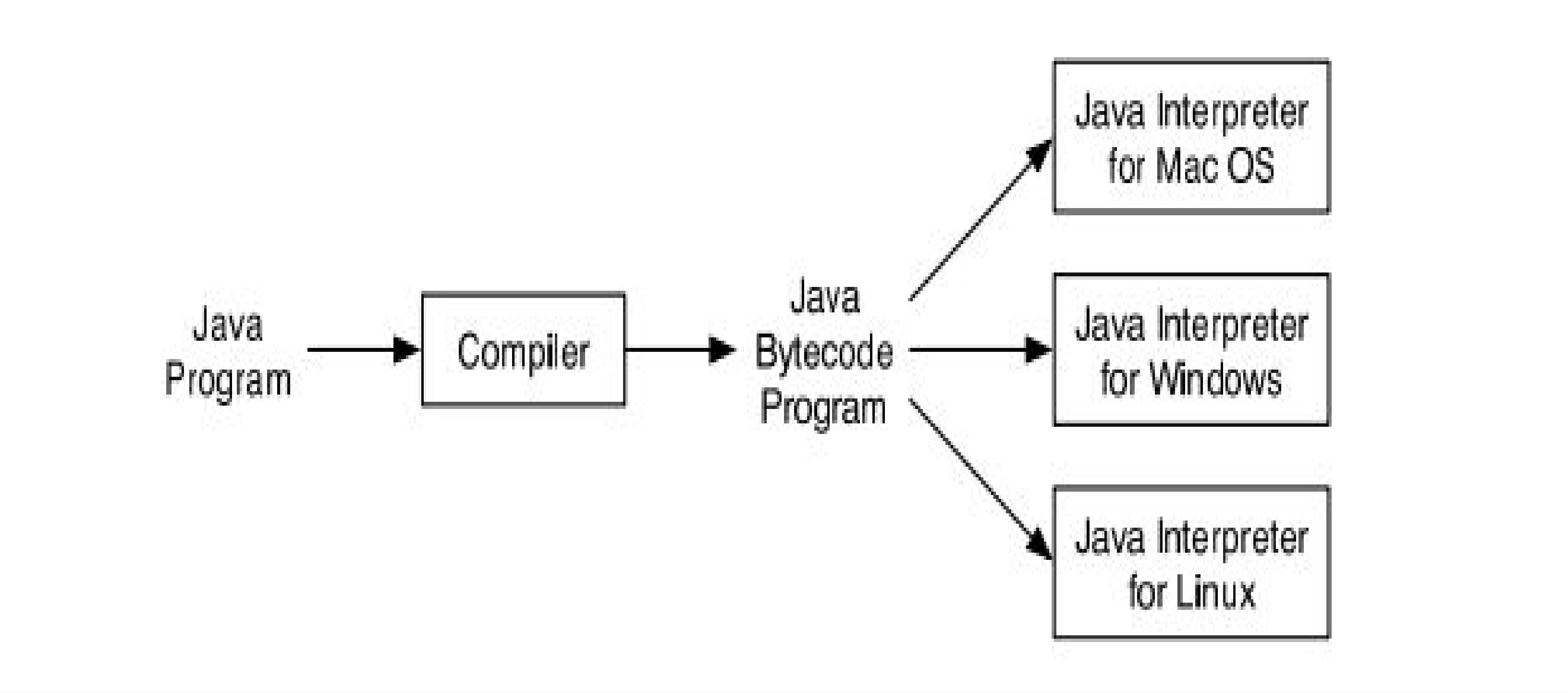
The Java Virtual Machine

- Programs written in Java are compiled into machine language, but it is a machine language for a computer that doesn’t really exist.
- This so-called “virtual” computer is known as the Java virtual machine.
- The machine language for the Java virtual machine is called Java bytecode
- There is no reason why Java bytecode could not be used as the machine language of a
- real computer, rather than a virtual computer.

Java bytecode interpreter

- One of the main selling points of Java is that it can actually be used on any computer
- All that the computer needs is an interpreter for Java bytecode
- Such an interpreter simulates the Java virtual machine in the same way that Virtual PC simulates a PC computer
- Once a computer has a Java bytecode interpreter, it can run any Java bytecode program.

Java Compiler & Interpreter


Java Program
Compiler
Java Bytecode Program
Java Interpreter for Mac OS
Java Interpreter for Windows
Java Interpreter for Linux

Agenda

- Machine Language
- Asynchronous Events
- The Java Virtual Machine
- Building Blocks of Programs
- Object-oriented Programming
- The Modern User Interface
- The Internet

Building Blocks of Programs

- Two aspects of programming
  - Data and instructions
- To work with data
  - Need to understand variables and types
- To work with instructions
  - Need to understand control structures and subroutines

Control Structures

- Special instructions that can change the flow of control
- There are two basic types of control structure
  - Loops: allow a sequence of instructions to be repeated over and over
  - Branches: allow the computer to decide between two or more different courses of action by testing conditions that occur as the program is running

Subroutines

- Large programs should be divided into manageable ”chunks”
- A subroutine or subprogram (also called procedure, method, function, or routine) is a portion of code within a larger program
  - Performs a specific task and is relatively independent of the remaining code
- Subroutines, often collected into libraries, are an important mechanism for sharing and trading software

Agenda

- Machine Language
- Asynchronous Events
- The Java Virtual Machine
- Building Blocks of Programs
- Object-oriented Programming
- The Modern User Interface
- The Internet

Structured Programming

- To solve a large problem, break the problem into several pieces and work on each piece separately
- This approach is called top-down programming
- It deals almost entirely with producing the instructions necessary to solve a problem
- Top-down programming doesn't give adequate consideration to the data that the program manipulates
- It makes it difficult to reuse work done for other projects

Bottom-up Design

- The approach is to start ”at the bottom” with problems that you already know how to solve
- and a reusable software component to solve that problem
- From reusable software components, you can work upwards a solution to the overall problem
- The reusable components should be as ” modular” as possible

Module

- A module is a component of a larger system that interacts with the rest of the system
- The idea is that a module can be ”plugged into” a system
- The details of what goes on inside the module are not important to the system as a whole, as long as the module fulfills its assigned role correctly
- This is called information hiding which is one of the most important

Object-oriented Programming

- Object
  - A kind of module containing data and subroutines
  - A kind of self-sufficient entity that has an internal **state** (the data it contains) and that can respond to **messages** (call to its subroutines)
- Example:
  - A mailing list of object has a state consisting of a list of names and addresses
  - If you send it a message telling to add a name, it will respond by modifying its state to reflect the change

The OOP Approach

- Identifying the objects involved in a problem and the messages that those objects should respond to
- The program
  - A collection of objects, each with its own data and its own set of responsibilities
  - The objects interact by sending messages to each other

Polymorphism

- You should think of objects as ”knowing” how to respond to certain messages
  - Example: a ”print” message would produce very different results, depending on the object it is sent to
- Polymorphism: Different objects can response to the same message in different ways

Objects and Classes

- Objects that contain the same type of data and that respond to the same messages in the same way belong to the same **class**
- A subclass is a class that inherits some properties from its superclass
- You can usually think of the subclass as being "a kind of" its superclass

Example of Subclass

- A square is a kind of a rectangle
- A rectangle has four sides with lengths w and h
- A square has all of the characteristics of a rectangle; in addition, w = h

Objects and Instances

- In a language where each object is created from a class, an object is called an instance of that class
- If each object has a type, two objects with the same class would have the same data type
- Creating an instance of a class is sometimes referred to as instantiating the class
- Example:
  - Square s = new Square();

// Square is a class, s is an object of class Square

Inheritance

- Inheritance is a way to form new classes using classes that have already been defined
- The new classes, known as derived classes, take over (or inherit) attributes and behavior of the pre-existing classes, which are referred to as base classes (or ancestor classes)
- It is intended to help reuse existing code with little or no modification.

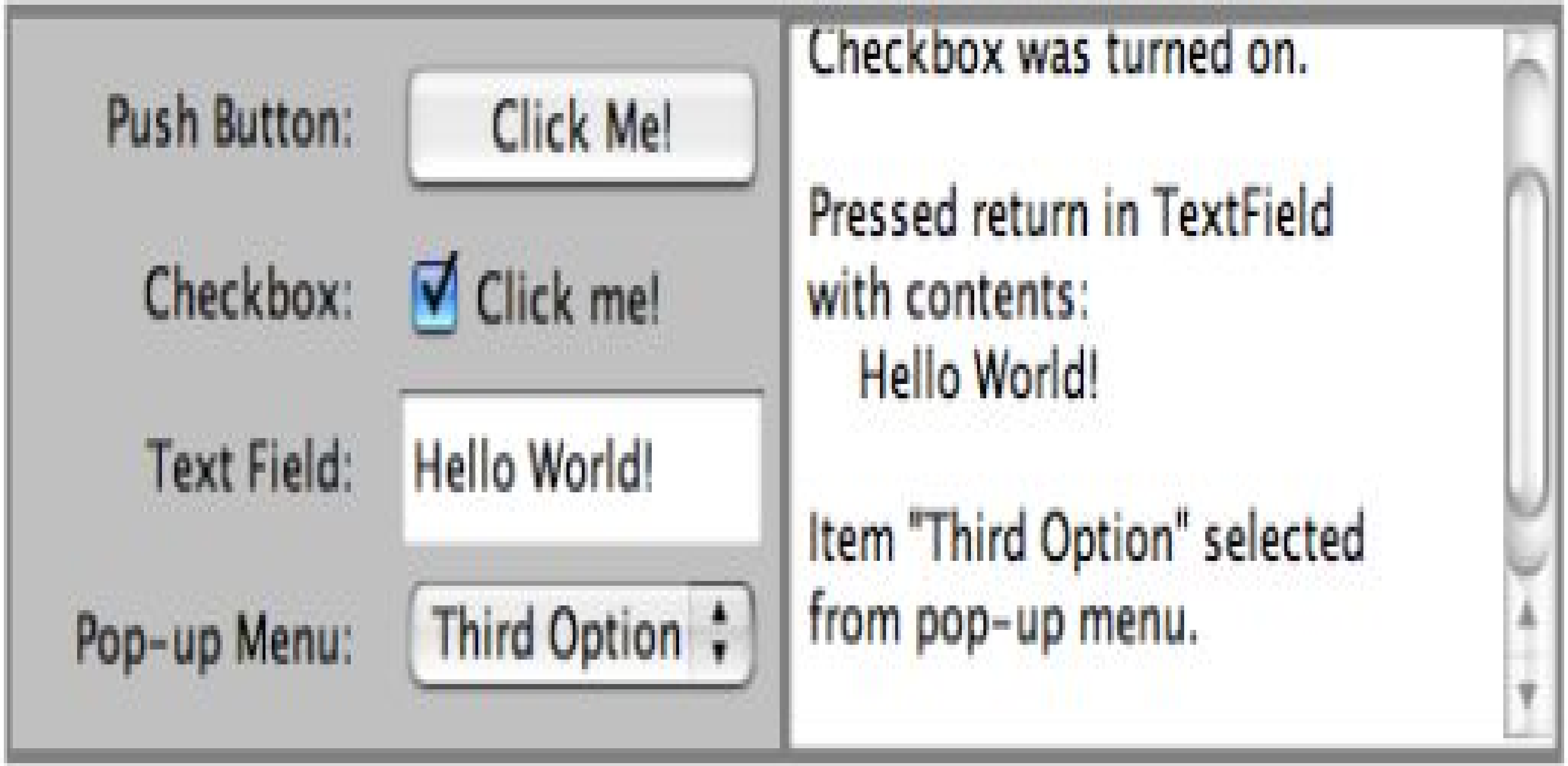
Agenda

- Machine Language
- Asynchronous Events
- The Java Virtual Machine
- Building Blocks of Programs
- Object-oriented Programming
- The Modern User Interface
- The Internet

User Interface

- Command-line interface
  - A user type commands to the computer via the terminals and the computer type
- Graphical User Interface (GUI)
  - The computer draws interface components on the screen, such as windows, scroll bars, menus, buttons, and icons
  - Usually, a mouse is used to manipulate such components

A simple applet



```
Checkbox was turned on.

Pressed return in TextField
with contents:
    Hello World!

Item "Third Option" selected
from pop-up menu.
```

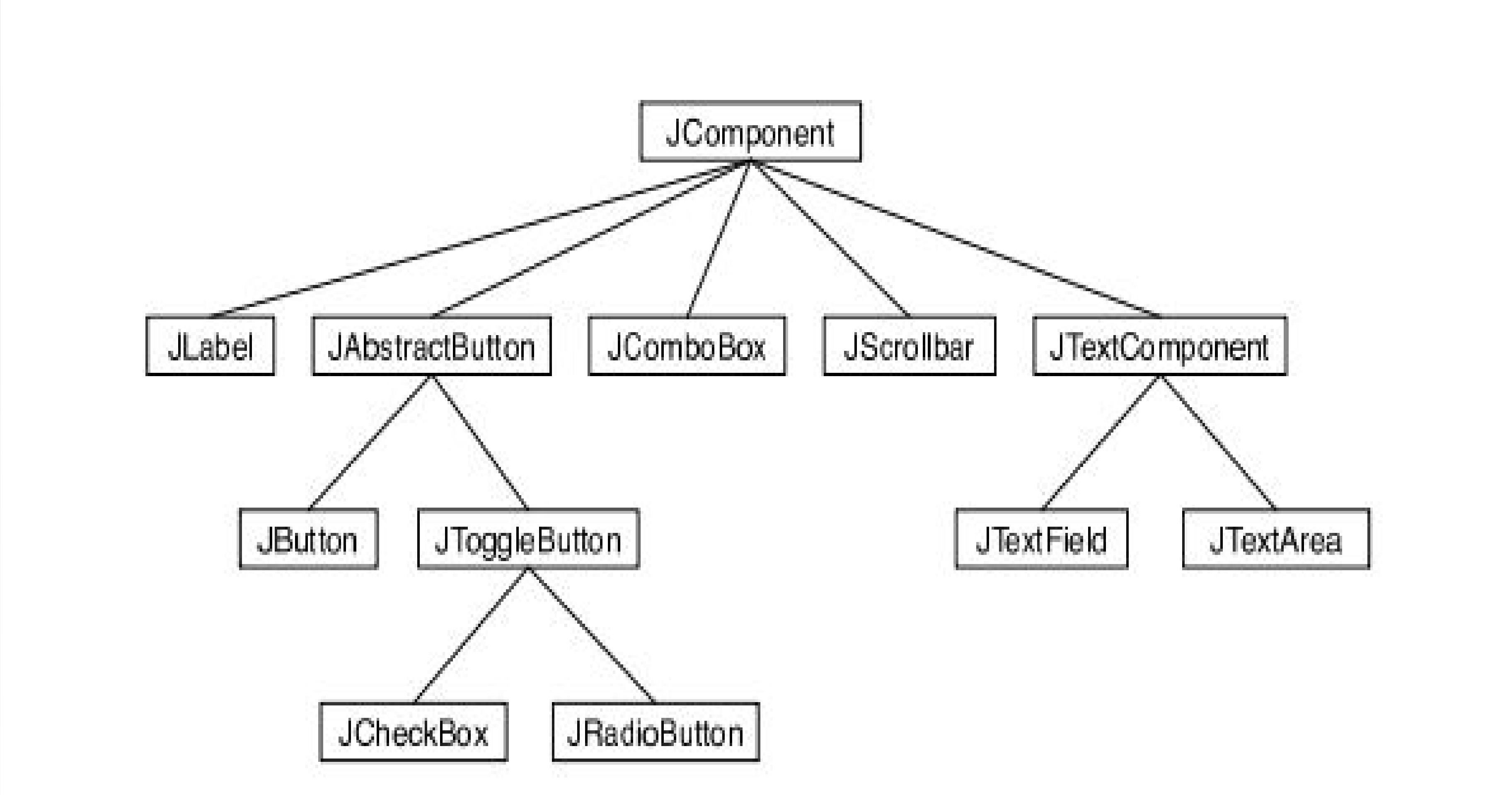
Java sets of GUI components

- The AWT or Abstract Windowing Toolkit
  - Java's original platform-independent windowing, graphics, and user-interface widget toolkit
- Swing
  - Swing is used in preference to the
  - AWT in most modern Java programs
  - Swing relies on AWT for its interface to the native windowing system.

User Interaction: Event

- When a user interacts with the GUI components, an ”event” is generated
- Each time an event is generated, a message is sent to the applet telling it that the event has occurred
- The program consists of ”event handlers” that tell the applet how to respond to various types of events

Swing's GUI classes



Agenda

- Machine Language
- Asynchronous Events
- The Java Virtual Machine
- Building Blocks of Programs
- Object-oriented Programming
- The Modern User Interface
- The Internet

The Internet and the WWW

- Computers can be connected together on networks
- Millions of computers throughtout the world are connected to a single huge network called the **Internet**
- There are elaborate protocols for communication over the Internet
  - A protocol is simply a detailed specification of how communication is to proceed

IP & TCP

- The most basic protocols on the Internet are IP and TCP
- Internet Protocol (IP) specifies how data is to be physically transmitted from one computer to another
- Transmission Control Protocol (TCP) ensures that data sent using IP is received in its entirety and without error

Packets

- All communication over the Internet is in the form of packets
- A packet consists of some data being sent from one computer to another, along with addressing information
- Think of a packet as an envelope with an address on the outside and a message on the inside
- Long messages must be divided among several packets

IP Address

- Every computer on the Internet has an IP address
- An IP address is a number that identifies it uniquely among all the computers on the net
- The IP address is used for addressing packets
- A computer can only send data to another computer if it knows that computer's IP address

Domain Names

- Since people prefer to use names rather than numbers, most computers are also identified by names, called domain names
- Sample domain name: gear.kku.ac.th
- Domain names are just for convenience
- Your computer still needs to know IP addresses before it can communicate
  - Using a domain name server to find out the corresponding IP address

Internet Services

- The Internet provides a number of services to the computers connected to it
- These services use TCP/IP to send various types of data over the net
- The most popular services
  - Instant messaging
  - File sharing
  - Electronic mail
  - The World Wide Web

Email Service

- The email service uses a protocol known as SMTP (Simple Mail Transfer protocol) to transfer email messages
- Other protocols, such as POP (Post Office Protocol) and IMAP (Internet Message Access Protocol) are used to fetch messages from an email
- A person who use emails does not need to understand these protocols
- The email programs provide an easy-to-use user interface to the underlying protocols

The World-Wide Web

- The World-Wide Web is perhaps the most exciting of network services
- The WWW allows you to request **pages** of information that are stored on computers all over the Internet
- A Web page contain **links** to other Web pages
- A computer that stores such pages of information is called a **web server**
- The user interface to the Web is the type of program known as a **web browser**

Web Browser

- You use a web browser to request a page of information
- The browser will send a request for that page to the web server on which the page is stored
- When a response is received from that web server, the web browser displays it to you in a neatly formatted form
- A web browser is just a user interface to the Web

HTTP

- Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems
- Its use for retrieving inter-linked resources led to the establishment of the World Wide Web
- HTTP is a request/response standard of a client and a server
  - A client is the browser of the end-user
  - The server is the web server

Java and the Internet

- Special java programs called applets are meant to be transmitted over the Internet and display on Web pages
- A web browser that understands Java can run the applet right on the Web page
- Many of the largest and most complex web sites use web server software that is written in Java

Summary

- The component of a computer that processes data is a CPU
- A CPU works with memory to run the program which needs to be translated into a machine language
- OOP has been adopted in software engineering to efficiently produce high quality software
- A modern computer has Internet access and has been equipped with GUI software

Questions

- Explain what is meant by an “asynchronous event.”
- What is the difference between a “compiler” and an “interpreter”
- Java is an object-oriented programming language. What is an object?
- What is the ”Internet”? Give some examples of how it is used

References

- David J. Eck, ”Introduction to Programming Using Java”, Version 5.0, December 2006