Computers Are Your Future



© 2006 Prentice Hall, Inc.

Computers Are Your Future

Week8 Lecture 3 Chapter 11

Programming Languages and Program Development

© 2006 Prentice Hall, Inc.



Java

✓ Java:

- Developed by Sun Microsystems
- An object-oriented, high-level programming language with a twist
- First true cross-platform programming language
- Gained acceptance faster than any other programming language
- > A simplified version of C++

Java

- ✓ Java, continued :
 - > Java is designed to run on any computer platform
 - Java Virtual Machine enables cross-platform use
 - Java applets or small programs are downloaded to computers through networks
 - > Weaknesses include:
 - The security risk in downloading applets
 - The speed in running the programs

Computers Are Your Future: Chapter 11 Sample Java Program

```
public void ejbCreate(String person, String id)
   throws CreateException {
   if (person == null) {
      throw new CreateException("Null person not allowed.");
   }
   else {
      customerName = person;
   }
   IdVerifier idChecker = new IdVerifier();
   if (idChecker.validate(id)) {
      customerId = id;
   3
   else {
      throw new CreateException("Invalid id: "+ id);
   }
   contents = new Vector();
3
```



Web-Based Languages

- ✓ Markup languages:
 - Hypertext markup language (HTML) sets the attributes of text and objects within a Web page
 - Extensible markup language (XML) is used for sharing data and objects in a Web environment
- ✓ Scripting languages:
 - VBScript is used to write short programs (scripts) that are embedded in Web pages
 - JavaScript is used to write scripts on Web pages
- ✓ Visual Studio .NET:
 - Used for the development of scripts and programs that are accessible from the Web



The Program Development Life Cycle (PDLC)

- ✓ The PDLC was introduced in the 1970s to address problems in creating programs
- ✓ It provides an organized plan for breaking down the task of program development into manageable parts
- \checkmark Six phases of the PDLC:
 - 1. Defining the problem
 - 2. Designing the program
 - 3. Coding the program
 - 4. Testing and debugging the program
 - 5. Formalizing the solution
 - 6. Implementing and maintaining the program



Phase 1: Defining the Problem

- \checkmark The first step in program development
- ✓ Systems analysts provide program specifications (specs) to programmers
- ✓ Specs define:
 - Input data
 - > Processing
 - > Output
 - > Appearance of user interface



Computers Are Your Future: Chapter 11 Phase 2: Designing the Program

- \checkmark Programmers create the program's design
 - Top-down design focuses on the program's main goal (main routine), then breaks the program into manageable components (subroutines/modules)
 - Control structures are used to see how each subroutine will do its job
- ✓ Developing an algorithm is a step-by-step description of how to arrive at a solution
- ✓ Program design tools:
 - Structure charts show the top-down design
 - Flow charts show the logic of program
 - Pseudo code alternative to flow charts



Computers Are Your Future: Chapter 11 Structured Design

- ✓ **Control structures** are logical constructs that specify how the instructions in a program are to be executed
- ✓ Three types of control structures:
 - Sequence control structure Instructions are executed in the order in which they appear
 - Selection control structures The program branches to different instructions depending on whether a condition is met; IF...THEN...ELSE
 - Repetition control structure The program repeats the same instructions over and over; DO-WHILE and DO-UNTIL



Phase 3: Coding the Program

✓ Coding requires the translation of the algorithm into specific program instructions

✓ An appropriate programming language is chosen, and the code is typed according to its syntax rules



Phase 4: Testing and Debugging the Program

- \checkmark Testing and debugging eliminate all errors
- \checkmark Syntax and logic errors are corrected
- \checkmark Debugging is the process of eliminating errors



Phase 5: Formalizing the Solution

- \checkmark Documentation is created for future use
- ✓ The variable names and definitions, a description of the files needed, and the layout of the output are produced
- ✓ A user manual is developed to explain how the program works



Phase 6: Implementing and Maintaining the Program

The program is:
 Tested by users
 Thoroughly documented
 Maintained and evaluated regularly

Chapter 12 Summary

- A programming language is an artificial language consisting of a vocabulary and a set of rules
- Machine language is the lowest-level programming language
- Assembly language contains symbols for programming instructions
- Third-generation (high-level) languages require programmers to specify the procedures to be followed
- Object-oriented languages combine procedures and data



Chapter 12 Summary, continued

- The PDLC's six phases are:
 - Defining the program
 - Designing the program
 - Coding the program
 - Testing and debugging the program
 - Formalizing the solution
 - Implementing and maintaining the program
- Top-down programming makes programs easier to debug and maintain
- Debugging requires finding and correcting syntax errors and logic errors

