Introduction and Background

Object oriented programming

What Is a Computer?

- An electronic device that executes the instructions in a program.
  
  Output: produces output  
  Processor: processes data  
  (CPU: central processing unit)

  Input: accepts data  
  Memory: stores information

- First electronic computer invented during 1930s and 1940s.
Overview of Computer Science

A.I.
- Intelligent agents
- Multi-agent systems
- Computer vision
- Nature language processing
- Robert

Theory
- Algorithm
- Discrete Math.
- Computational theory

System
- Architecture
- Compiler
- Operating system
- Database
- Network

Application
- Interface
- Software engineering
- Programming language

E-commerce; Bio-Informatics; Information retrieval; Computational physics; Computational chemistry; Medical image processing;

To Be a Successful Computer Scientist

- You will need:
  - Mathematics and logic reasoning as foundation
  - Rigorous practice as engineer
  - Creative thinking as artist
  - Knowledge from multiple disciplines
  - Learn new ideas quickly and not afraid of fast-changing world
  - Follow ethical and moral rules
Binary Representation

- Numbers written with just 0 and 1, are called **binary** numbers.
- Each 1 is a power of 2 so that the digits in the figure represent the number:
  \[ 13 = 2^3 + 0 + 2^5 + 0 + 2^3 + 2^2 + 0 + 0 \]
  \[ = 128 + 0 + 32 + 0 + 8 + 4 + 0 + 0 \]
  \[ = 172 \]

Practice

- Write down the binary representation for “13”
- $13 \div 2 = 6$ remainder $= 1$
- $6 \div 2 = 3$ remainder $= 0$
- $3 \div 2 = 1$ remainder $= 1$
- $1 \div 2 = 0$ remainder $= 1$
- The answer is: 1101
Machine Language

- The native tongue of the CPU
- Each design of CPU has its own machine language
- Made up of sets of 0’s and 1’s
- Very difficult for people to use

The segment of Java code:

```
int counter = 0;
Counter = counter + 1;
```

Might be translated into machine language as:

```
00010100010001000100010001000100010101011110
00000111101010001110000100010000010101010
```

Assembly Language

- Code or abbreviations for machine language instructions
  - `ADD A B C; C contains A+B`
  - `MUL C D; now D contains (A+B)*(A+B)`
- Assembler: the program which translates assembly language to machine language
Some High Level Languages

- **FORTRAN** = formula translation
  - Created around 1957 for scientific computation
  - To describe complex situations, like space flight, nuclear explosions
    
    ```fortran
    Program idexmp
    Integer bus_num
    Bus_num = 99
    Write(*,*) 'take bus ', bus_num
    End
    ```

Some High Level Languages - Continue

- **COBEL** = common business oriented language.
  - Developed about 1960, for business application.
- **BASIC** = beginner's all symbolic instruction code.
  - Written in 1964, for college students to use to learn programming concepts.
  - Looks like English.
- **C, c++**.
  - Originally created for writing system software.
  - Widely used by programming professionals for all sorts of programs.
Translator

- From source code in high-level language to object code in machine language
- Two types of translators
  - Interpreter: translates a single source code statement into it machine code, execute it, and then goes on to the next source code statement
  - Compiler: translate the entire source code file and generates an object code file than then be executed

Overview of Languages

- Machine languages The lowest level language. Composed of 0’s and 1’s
- Assembly languages abbreviations for machine language
- High-level languages use program statements - words and algebra-type expressions
- 4th generation languages = 4GL. Very high-level, results oriented. I.E. Database query languages
  - Find student (course contains “CIS180” and “major” ! “CIS”)
- Natural languages 5th generation languages. "Who are the salesmen with sales over $20,000 last month?"
Programming

- What task? - What the program suppose to do?
  - Meet with user, get problem specification
  - Analyze the problem
  - Identify problem components
- What language?
  - You got already:
    - Existing standard; existing hardware; existing software
    - Programmer’s current knowledge
  - In the future
    - With variety of hardware
    - Changes easy to make in program, error easy to find in program

Different Languages

Languages: C, Java, C++, Fortran, Visual Basic, XBase, Powerbuilder, JavaScript, Word Basic, etc.

used for:
- System software
- Applications and Components
- Applications
- Scripts & Macros
Programming - Who’s Involved

- Different people:
  - End user
  - System analyst - design the requirements and set the strategy for program
  - Programmer - write the actual code
- Communication back and forth
- A highly structured series of conferences and consumer surveys
- Costs come into play

Program Development

- Set & review goals: what is it supposed to do?
- Design: create the strategy to achieve goal.
- Coding: write the program.
- Testing: try it out with real people, different conditions.
- Documentation: what you did and why. How to use it.
Flowchart

- Different shapes for different kinds of steps, like input and output, decisions, calculations.
- Keep logic clear

Programming Paradigms - 1

- Unstructured programming
  - Small and simple program contains only one main program
  - Main program: a sequence of statements that modify data which in global throughout whole program
  - Repeat use of the same statements requires copying program

main program
data
Programming Paradigms - 2

- Procedural programming
  - A single program is divided into procedures
  - Each procedure can be called (invoked) one or more times from the main program or other procedures
  - Easy to debug

![Diagram of main program and procedure relationships]

Programming Paradigm - 3

- Modular programming
  - Procedures of a common functionality are grouped together into separate module
  - Each module has its own data
  - Allow for team work - the implementation of each module can be done by different programmer

![Diagram of main program and modules relationships]
Programming Paradigm - 4

- Object oriented programming
  - A set of interacting objects
  - Each has its own data
  - Java is!

Internet and Worldwide Web

- Computer networks
  - A group of connecting computers
  - Share information and hardware
- Internet
  - A global network composed of smaller computer networks
- Client/server model
  - Server – computer that provide service
  - Client – computer that receives service
- Worldwide web (WWW)
  - A client/server application
  - WWW server stores web pages and transmits them to browser software (on client) such as NN and IE
HTTP Protocol

- HTTP: hypertext transfer protocol
  - A set of rules and conventions that govern communication between computers
- HTML: hypertext makeup language

![Diagram showing the HTTP Protocol](image)

Java and WWW

- Web pages were designed to be *device-independent* (*platform independent*).
- Java was too.
- Java was simple enough to.
  - Have a java interpreter embedded with a browser.
- Web page becomes alive with animation and user interactivity, using java.
Traditional Compiler Programs

- A program written in most programming languages, such as C, the compiler translates the program into machine code (executable program).
- The executable program is specific to the computer processor which it runs on.
- So, multiple executable programs needed for different systems.

Java Virtual Machine

- To have java program works: java compiler and java interpreter
Acknowledgement

- Some materials of this presentation are adopt from:
  - Dr emad aboelela's work

Information

- Development kit: eclipse
- Java virtual machine
  - Version after 1.3.0
Review Questions

- Write down the binary representation of “11”.
- Explain the difference between a compiler and an interpreter.
- What is “platform independent”? Why is it valuable?
- Explain the difference of “modular programming” and “objected oriented programming”?