Software Engineering

HUANG Jie
School of Computer Science and Technology
Tongji University

2025

Lesson 1
From Program to Software

- ✓ What is Program?
- ✓ What is Software?
- ✓ What is Software Engineering?
- ✓ What is the relationship between SEE and CS?
- ✓ Challenges in software development & evolution.
- ✓ Social responsibility of software engineer.

What is Program?

A Program is a set of statement sequences described by a programming language that can be understood and processed by a computer.

- Described in programming languages.
- Such as Java, C, C++, Python…

The program strictly follows the syntax and semantic rules of the programming language.

- Ensure that the program code can be understood by the compiler of the programming language, and then compiled to generate corresponding executable code.

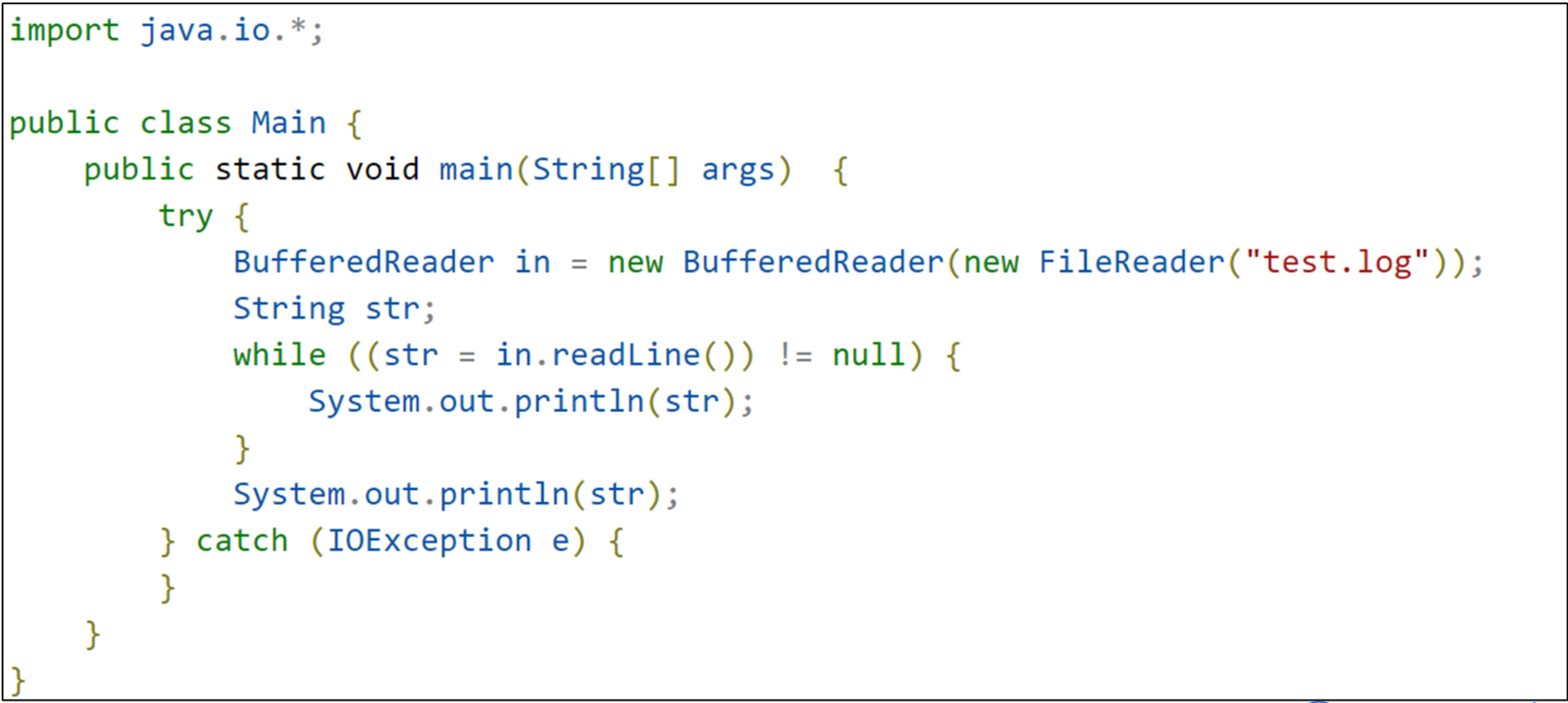
Program code can be presented in two forms:

- Source code: code described in a programming language
- Executable code: executable binary or intermediate code.

What is Program?

Example: Source code written in Java language

What function of this code?



```java
import java.io.*;

public class Main {
    public static void main(String[] args)  {
        try {
            BufferedReader in = new BufferedReader(new FileReader("test.log"));
            String str;
            while ((str = in.readLine()) != null) {
                System.out.println(str);
            }
            System.out.println(str);
        } catch (IOException e) {
        }
    }
}
```

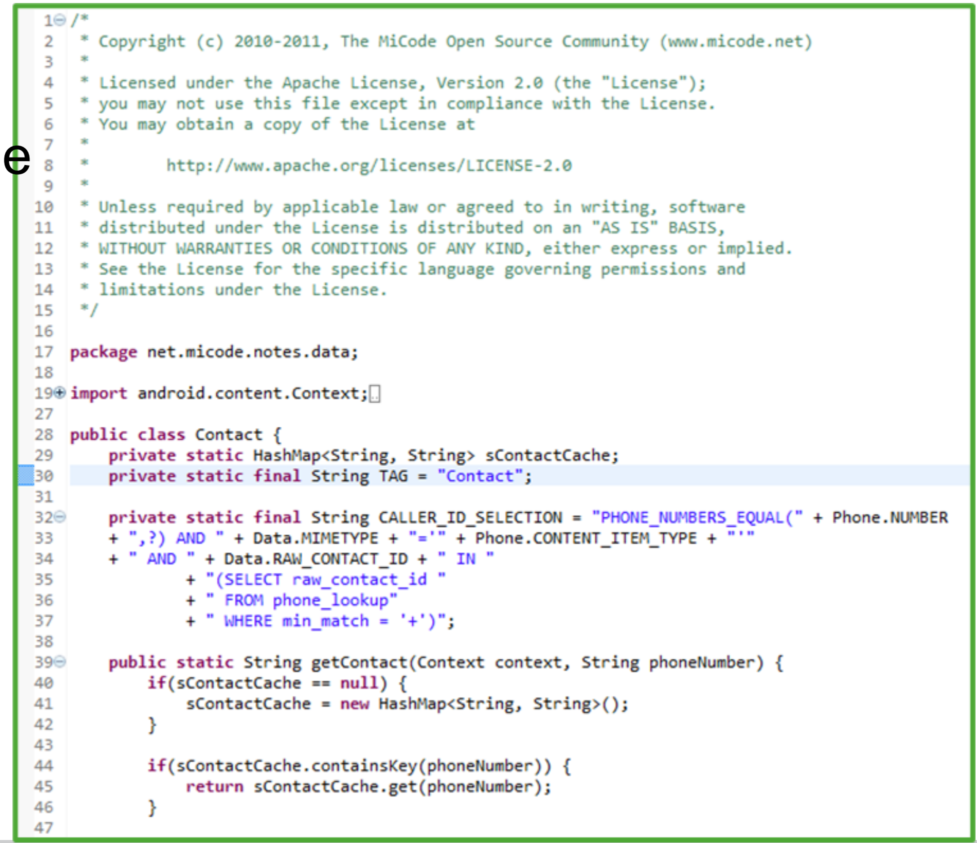
Program Composition - Statement

Statements in the program

- Declaration, definition, control, calculation, etc.
- Implement specific functions
- Description by using programming languages

Computers can understand

- Can be compiled into binary code
- Final executable



```java
/*
 * Copyright (c) 2010-2011, The MiCode Open Source Community (www.micode.net)
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *        http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package net.micode.notes.data;

import android.content.Context;

public class Contact {
    private static HashMap<String, String> sContactCache;
    private static final String TAG = "Contact";

    private static final String CALLER_ID_SELECTION = "PHONE_NUMBERS_EQUAL(" + Phone.NUMBER
    + ",?) AND " + Data.MIMETYPE + "='" + Phone.CONTENT_ITEM_TYPE + "'"
    + " AND " + Data.RAW_CONTACT_ID + " IN "
            + "(SELECT raw_contact_id "
            + " FROM phone_lookup"
            + " WHERE min_match = '+')";

    public static String getContact(Context context, String phoneNumber) {
        if(sContactCache == null) {
            sContactCache = new HashMap<String, String>();
        }

        if(sContactCache.containsKey(phoneNumber)) {
            return sContactCache.get(phoneNumber);
        }

```

Program Composition - Modules

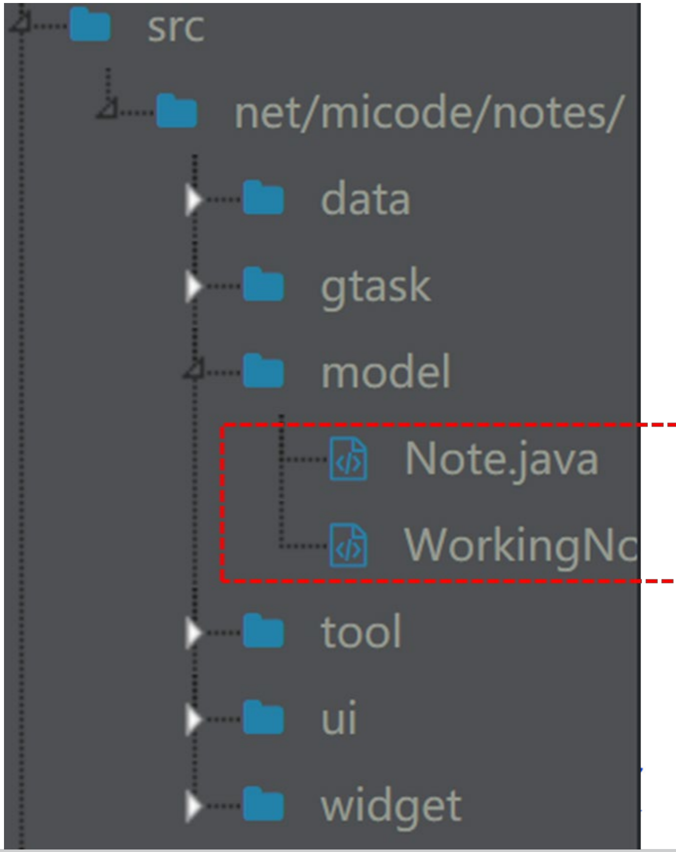
Composed of numerous interacting modules

- Package
- Class
- Method

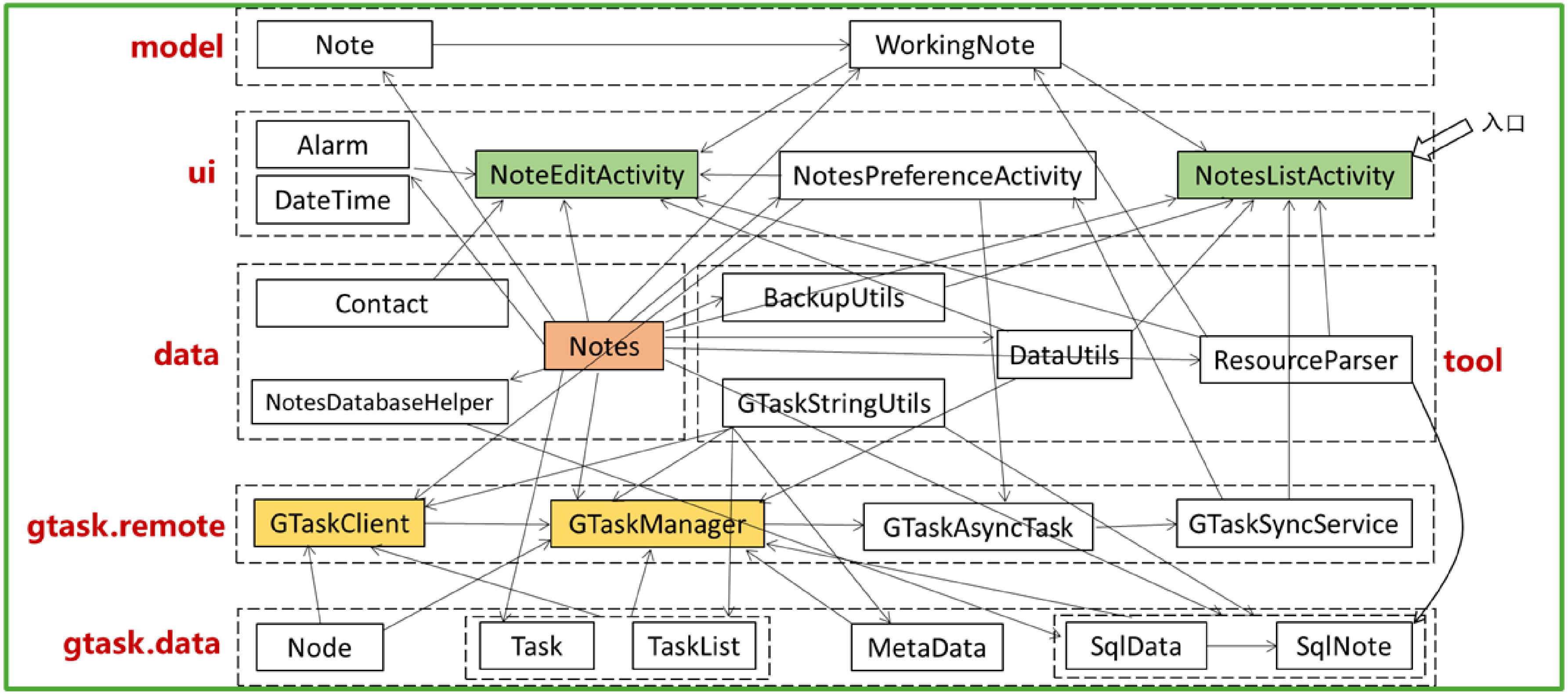
Each module implements specific functions

Example: Modules in Xiaomi Notes

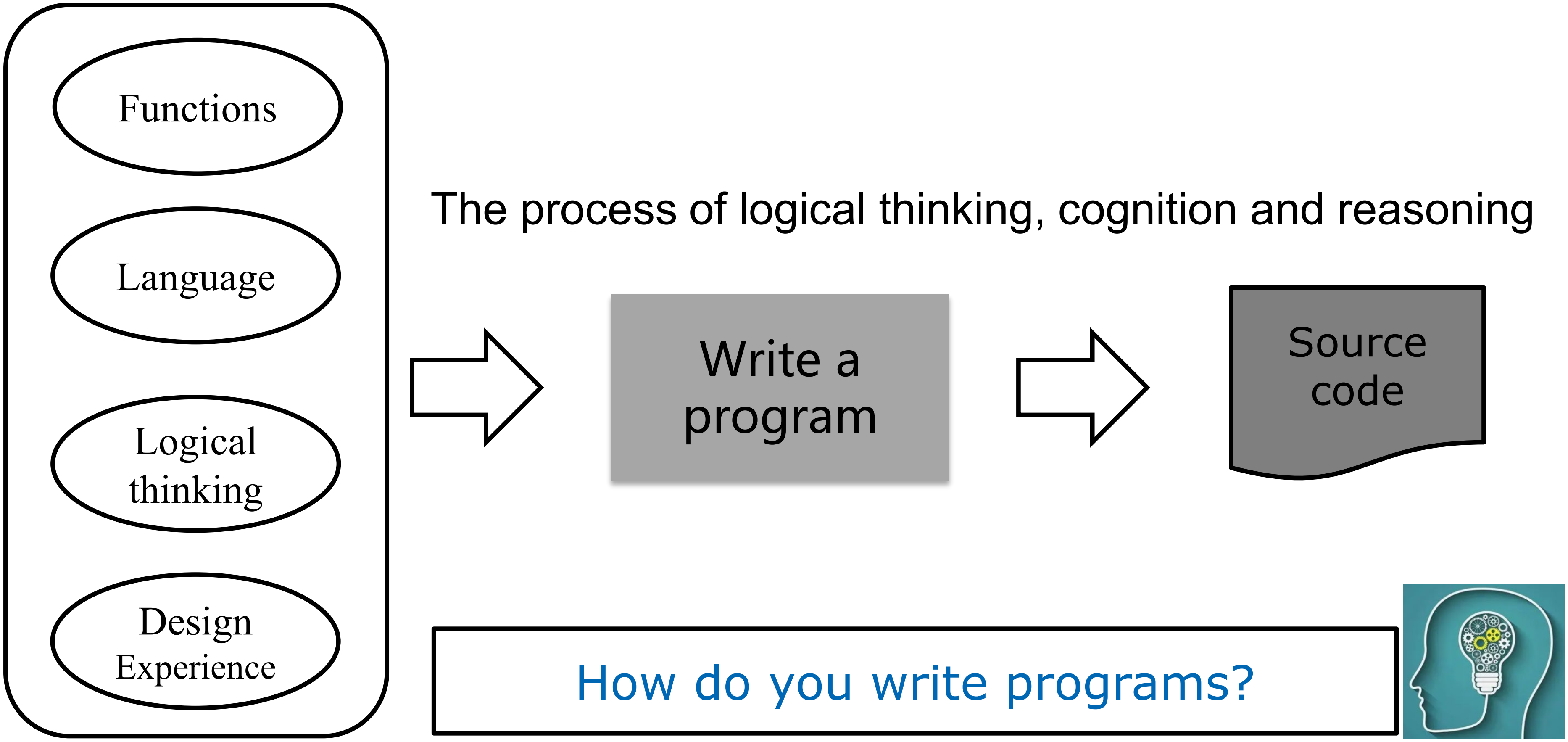
- 170 files
- 471 methods



Example: Module structure of Xiaomi Notes

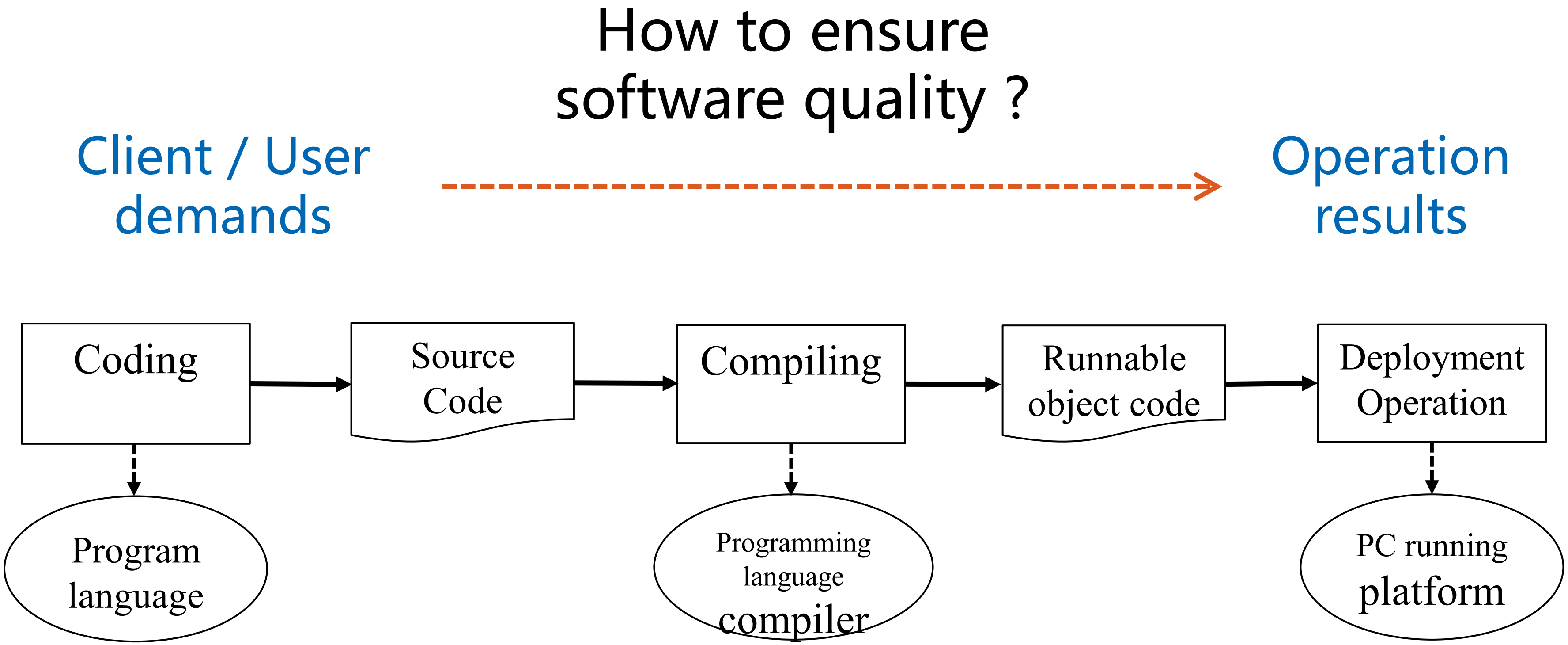


How to write a program?



How do you write programs?

Editing, compiling, deploying & running program



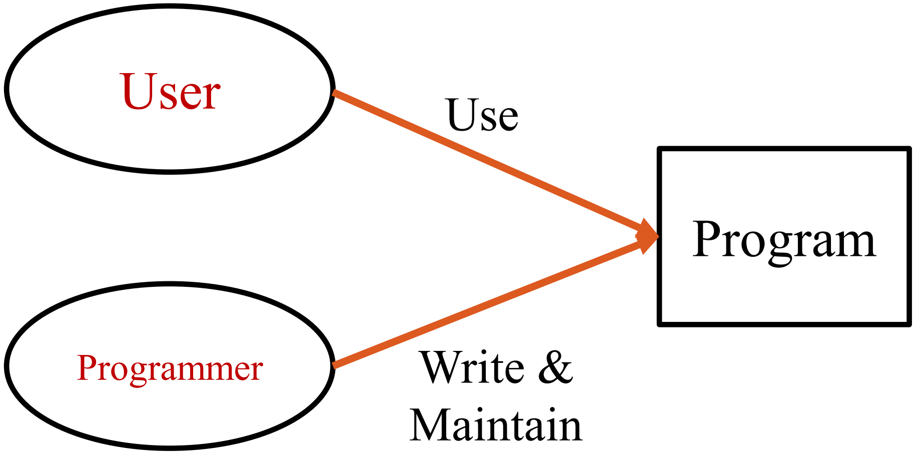
Stakeholders of the program

Client / User

- The running display function and performance of the program.
- Meet and fulfill the needs of users.

Programmer

- Write, read, and maintain programs.
- Identify and modify defects in the program.


User

Use

Program

Programmer

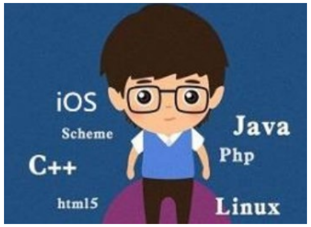
Write & Maintain

Different quality requirements for programs

User

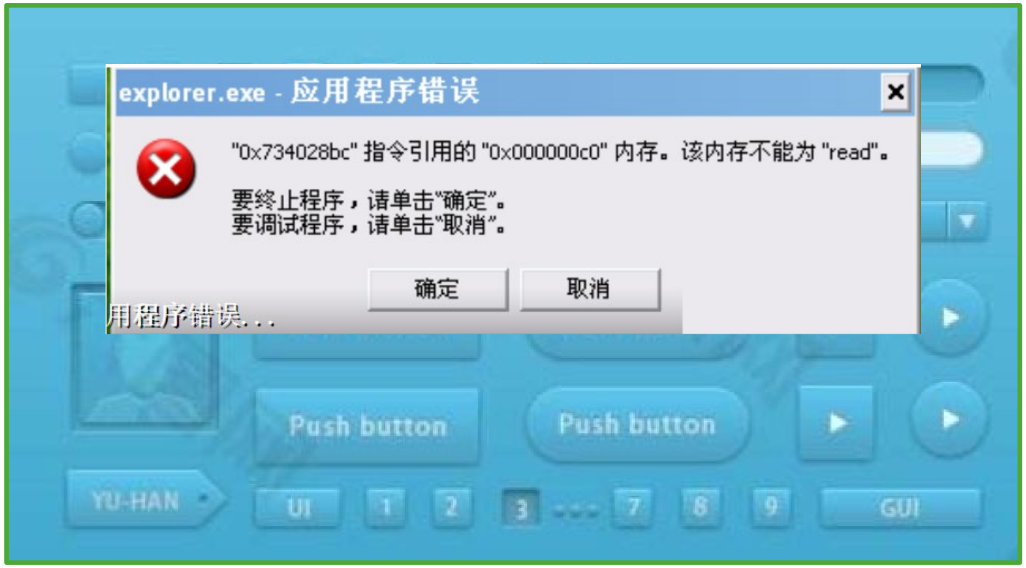
- Correctness
- Efficiency
- Reliability
- Friendliness
- ……

Programmer



- Understood
- Easy to modify
- Maintainable
- Reusable
- ……

Example: Program quality from the perspectives of users and programmers

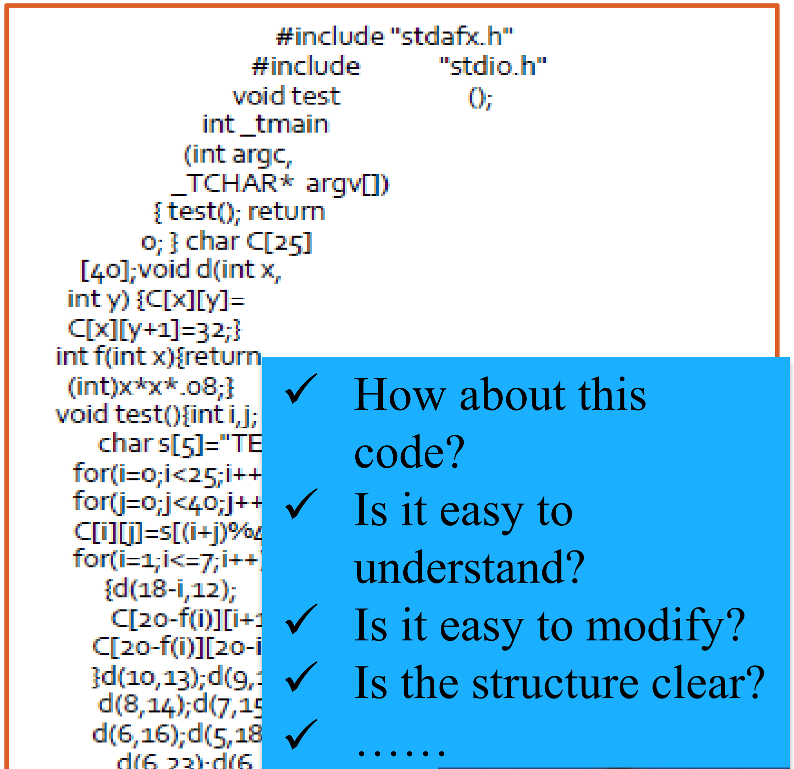


explorer.exe - 应用程序错误

"0x734028bc" 指令引用的 "0x000000c0" 内存。该内存不能为 "read"。

要终止程序，请单击"确定"。
要调试程序，请单击"取消"。

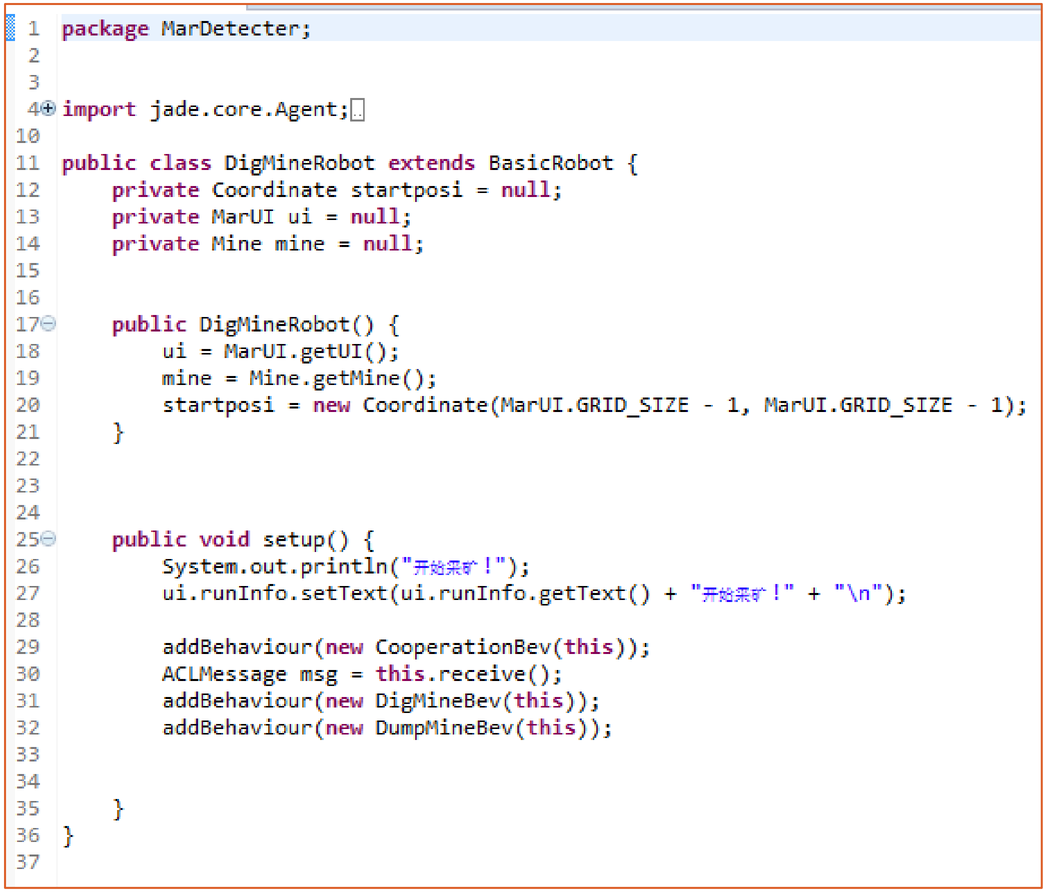
- ✓ How is the running effect?
- ✓ Is it reliable?
- ✓ Is it easy to operate and use?
- ✓ ……



```
#include "stdafx.h"
#include         "stdio.h"
void test          ();
int _tmain
(int argc,
_TCHAR* argv[])
{ test(); return
0; } char C[25]
[40];void d(int x,
int y) {C[x][y]=
C[x][y+1]=32;}
int f(int x){return
(int)x*x*.08;}
void test(){int i,j;
char s[5]="TE
for(i=0;i<25;i++
for(j=0;j<40;j++
C[i][j]=s[(i+j)%4
for(i=1;i<=7;i++)
{d(18-i,12);
C[20-f(i)][i+1
C[20-f(i)][20-i
}d(10,13);d(9,1
d(8,14);d(7,15
d(6,16);d(5,18
d(6,23);d(6,
for(j=0;j<40;printf("%c"
scanf("%c", &s[0]);
}
```

- ✓ How about this code?
- ✓ Is it easy to understand?
- ✓ Is it easy to modify?
- ✓ Is the structure clear?
- ✓ ……

Example: Quality of Program source code



```java
package MarDetecter;


import jade.core.Agent;

public class DigMineRobot extends BasicRobot {
    private Coordinate startposi = null;
    private MarUI ui = null;
    private Mine mine = null;


    public DigMineRobot() {
        ui = MarUI.getUI();
        mine = Mine.getMine();
        startposi = new Coordinate(MarUI.GRID_SIZE - 1, MarUI.GRID_SIZE - 1);
    }



    public void setup() {
        System.out.println("开始采矿！");
        ui.runInfo.setText(ui.runInfo.getText() + "开始采矿！" + "\n");

        addBehaviour(new CooperationBev(this));
        ACLMessage msg = this.receive();
        addBehaviour(new DigMineBev(this));
        addBehaviour(new DumpMineBev(this));


    }
}

```

How is this program written?

- ✓ Is it easy to understand?
- ✓ Is the structure clear
- ✓ Is it easy to modify?

……

Why?

……

Discussion in class

What kind of program do you think is of high quality?

Have you considered quality issues when writing programs?

How is the quality of the program you have written?

How can we write high-quality programs?

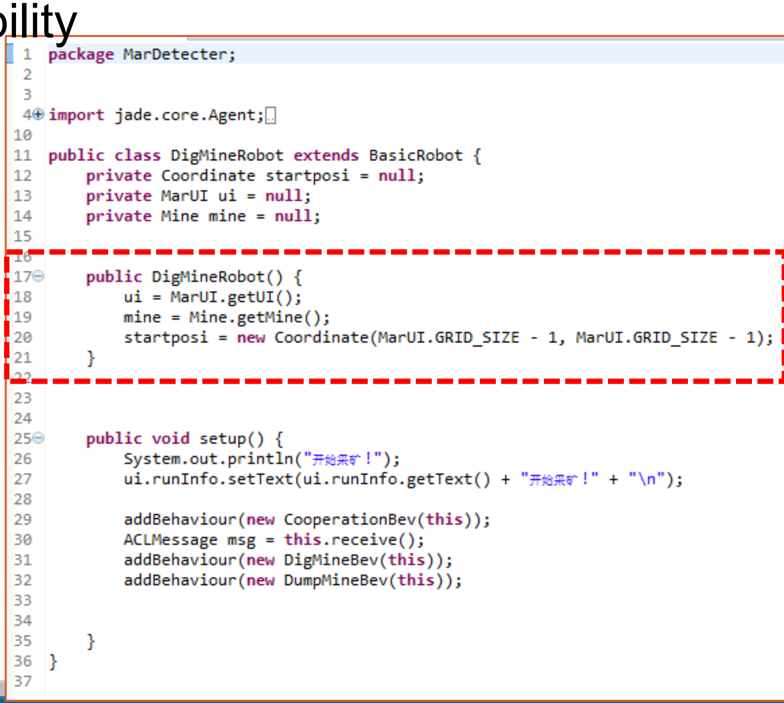
Program Quality Assurance Methods

Code Style Specification – Syntax

- Grammar and structural hierarchy
- Clarify how to standardize the writing of programs
- Displayed as whether it is easy to read and understand

Code Design Specification – Semantics

- Semantic and connotative(内涵的) levels, external
- Clarify how to organize and encapsulate program statements
- Displayed as good structure and easy reusability



```java
package MarDetecter;


import jade.core.Agent;

public class DigMineRobot extends BasicRobot {
    private Coordinate startposi = null;
    private MarUI ui = null;
    private Mine mine = null;


    public DigMineRobot() {
        ui = MarUI.getUI();
        mine = Mine.getMine();
        startposi = new Coordinate(MarUI.GRID_SIZE - 1, MarUI.GRID_SIZE - 1);
    }



    public void setup() {
        System.out.println("开始采矿！");
        ui.runInfo.setText(ui.runInfo.getText() + "开始采矿！" + "\n");

        addBehaviour(new CooperationBev(this));
        ACLMessage msg = this.receive();
        addBehaviour(new DigMineBev(this));
        addBehaviour(new DumpMineBev(this));


    }
}

```

Program Quality Assurance Methods

Follow coding style

Adopting programming methods

Carry out code reuse

Follow coding style

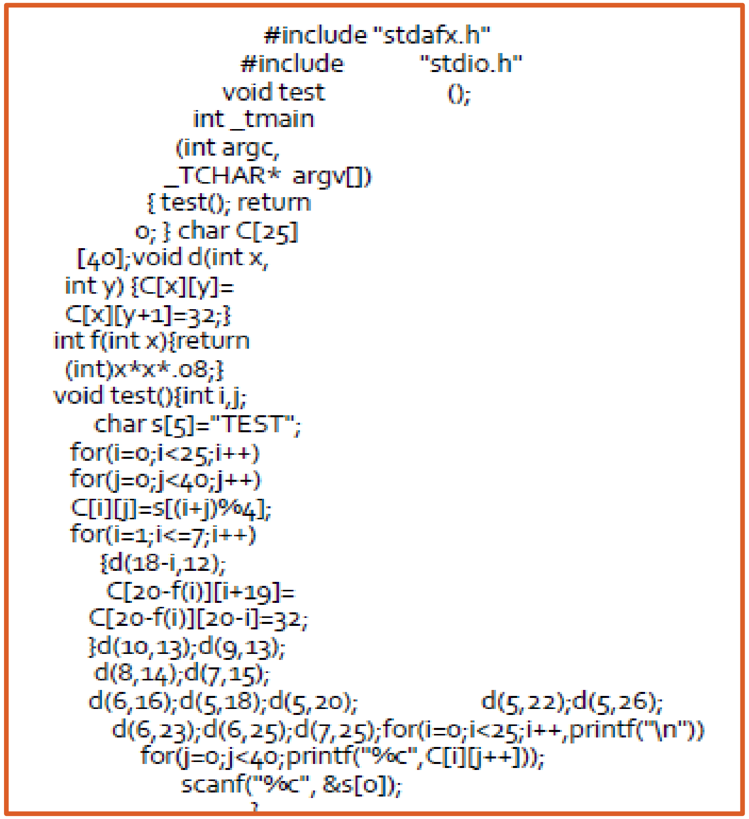
Good programming behavior

In the coding process, programmers organize code symbols well, name them reasonably, and provide necessary comments, which can enhance the readability and comprehensibility of the code, thereby improving its maintainability and reusability, and enhancing the internal quality of the code.

What is coding style?

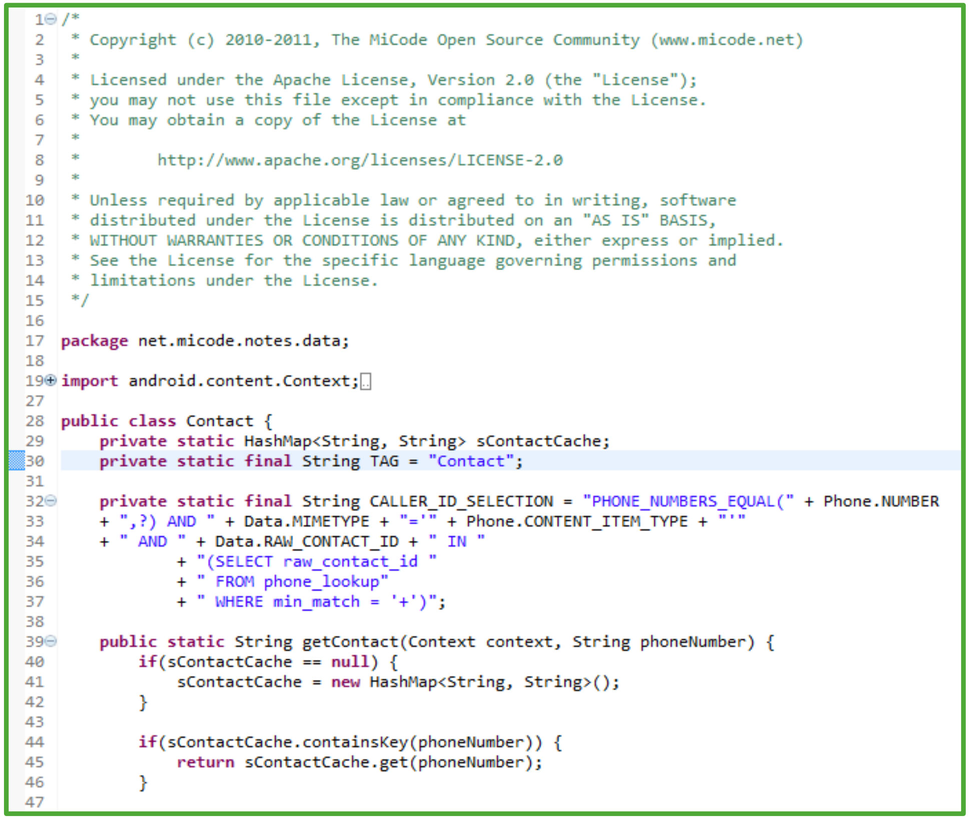
Programmers need to follow specific styles and requirements when coding to standardize their programming behavior and the style of the generated program code.

Follow coding style



```
                    #include "stdafx.h"
                  #include         "stdio.h"
                 void test              ();
              int _tmain
             (int argc,
             _TCHAR*  argv[])
           { test(); return
          0; } char C[25]
     [40];void d(int x,
    int y) {C[x][y]=
    C[x][y+1]=32;}
   int f(int x){return
    (int)x*x*.08;}
   void test(){int i,j;
       char s[5]="TEST";
     for(i=0;i<25;i++)
     for(j=0;j<40;j++)
     C[i][j]=s[(i+j)%4];
     for(i=1;i<=7;i++)
        {d(18-i,12);
         C[20-f(i)][i+19]=
       C[20-f(i)][20-i]=32;
       }d(10,13);d(9,13);
        d(8,14);d(7,15);
       d(6,16);d(5,18);d(5,20);            d(5,22);d(5,26);
         d(6,23);d(6,25);d(7,25);for(i=0;i<25;i++,printf("\n"))
            for(j=0;j<40;printf("%c",C[i][j++]));
                scanf("%c", &s[0]);
                  }
```

Not following coding style



```java
/*
 * Copyright (c) 2010-2011, The MiCode Open Source Community (www.micode.net)
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *        http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package net.micode.notes.data;

import android.content.Context;

public class Contact {
    private static HashMap<String, String> sContactCache;
    private static final String TAG = "Contact";

    private static final String CALLER_ID_SELECTION = "PHONE_NUMBERS_EQUAL(" + Phone.NUMBER
    + ",?) AND " + Data.MIMETYPE + "='" + Phone.CONTENT_ITEM_TYPE + "'"
    + " AND " + Data.RAW_CONTACT_ID + " IN "
            + "(SELECT raw_contact_id "
            + " FROM phone_lookup"
            + " WHERE min_match = '+')";

    public static String getContact(Context context, String phoneNumber) {
        if(sContactCache == null) {
            sContactCache = new HashMap<String, String>();
        }

        if(sContactCache.containsKey(phoneNumber)) {
            return sContactCache.get(phoneNumber);
        }

```

following coding style

Basic principles for writing code

Easy to read, understood at a glance

- Understand the meaning and intent of the code.
- Literal symbols, indentation and parentheses, code comments, following coding style.

Concise and reduce complexity

- Simplify by avoiding complexity, no need for goto statements, reduce nesting layers, and simplify algorithms.

Easy to modify and maintain

- Easy to modify program code and add new code.
- Encapsulation, parameterization, modularity, hiding, constant element.

No ambiguity or ambiguity

- Don't let people misunderstand.

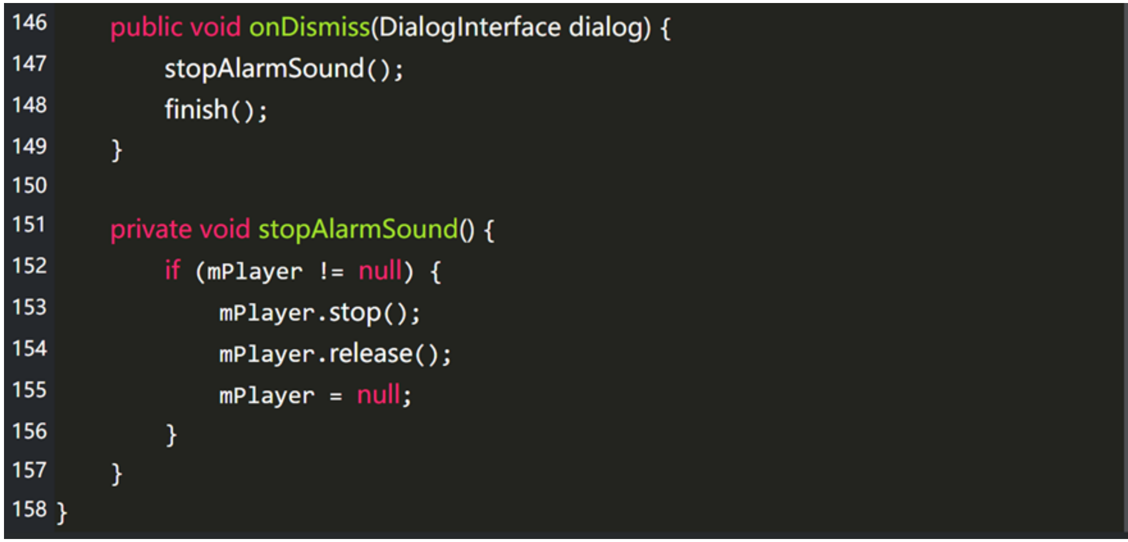
Coding Style - Code Layout

Indent(缩进), use the Tab key well.

Use parentheses to indicate priority.

{} at the break point

```
if (condition) {
        DoSomething();
} else {
        DoSomething();
}
```



```
    public void onDismiss(DialogInterface dialog) {
        stopAlarmSound();
        finish();
    }

    private void stopAlarmSound() {
        if (mPlayer != null) {
            mPlayer.stop();
            mPlayer.release();
            mPlayer = null;
        }
    }
}
```

At most one statement per line.

- Do not define multiple variables on one line.

Coding Style - Code Organization

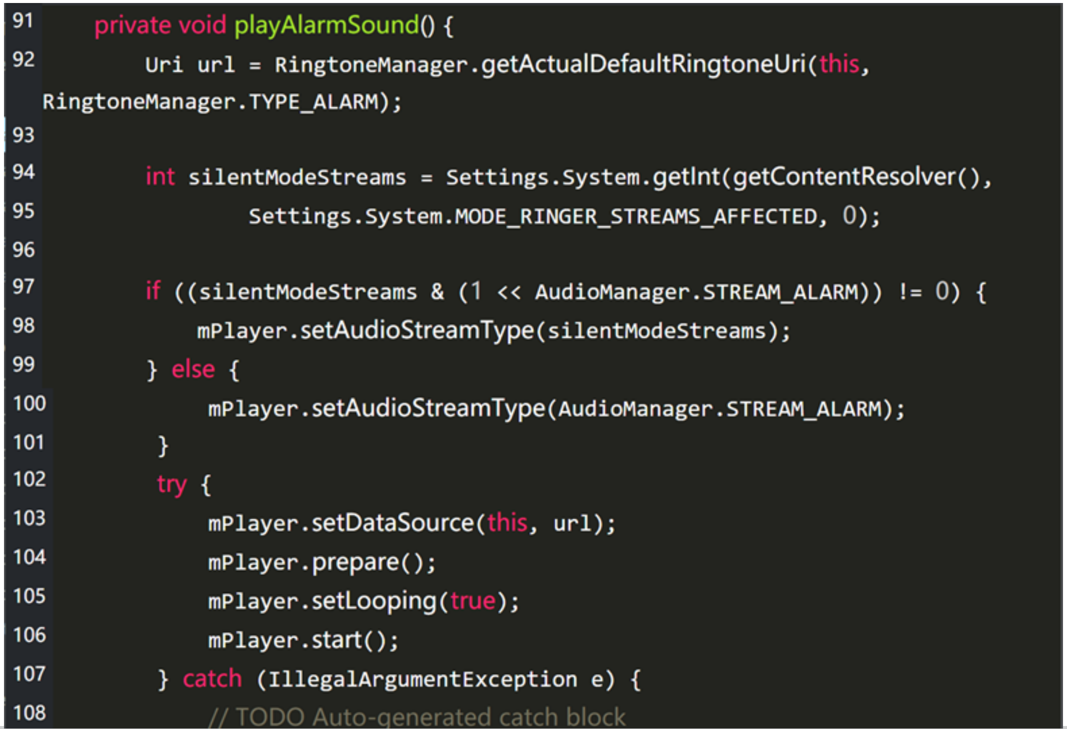
Explain the data in a certain order.

Explain objects in alphabetical order.

Try to avoid using nested(嵌套) structures as much as possible.

Adopt a unified indentation(缩进) rule.

Single entrance, single exit.



```
    private void playAlarmSound() {
        Uri url = RingtoneManager.getActualDefaultRingtoneUri(this,
RingtoneManager.TYPE_ALARM);

        int silentModeStreams = Settings.System.getInt(getContentResolver(),
                Settings.System.MODE_RINGER_STREAMS_AFFECTED, 0);

        if ((silentModeStreams & (1 << AudioManager.STREAM_ALARM)) != 0) {
            mPlayer.setAudioStreamType(silentModeStreams);
        } else {
            mPlayer.setAudioStreamType(AudioManager.STREAM_ALARM);
        }
        try {
            mPlayer.setDataSource(this, url);
            mPlayer.prepare();
            mPlayer.setLooping(true);
            mPlayer.start();
        } catch (IllegalArgumentException e) {
            // TODO Auto-generated catch block
```

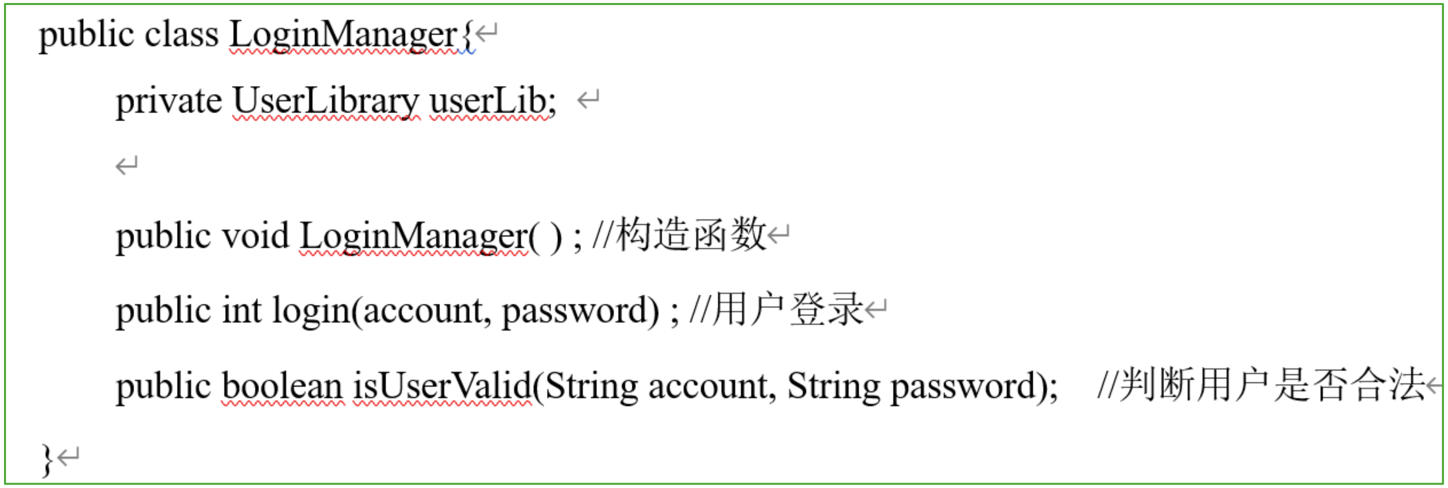
Coding Style - Naming conventions

Naming conventions

- Use English words or abbreviations, do not use PinYin.
- The principle of understanding the meaning through reading, with clear and distinct meanings.
- Naming should not be too long.
- Try to use the full name and avoid using abbreviations.

■ 不规范的命名

✓ DaYinWenJian 与 PrintFile



```
public class LoginManager{↵
        private UserLibrary userLib; ↵
        ↵
        public void LoginManager( ) ; //构造函数↵
        public int login(account, password) ; //用户登录↵
        public boolean isUserValid(String account, String password);    //判断用户是否合法↵
}↵
```

Coding Style - Code Annotations

Help understand the program

Annotations(注释) should explain the program:

- What it does?
- Why do you do this?
- Precautions(注意事项).
- No need to explain how the program works.

Annotation position：

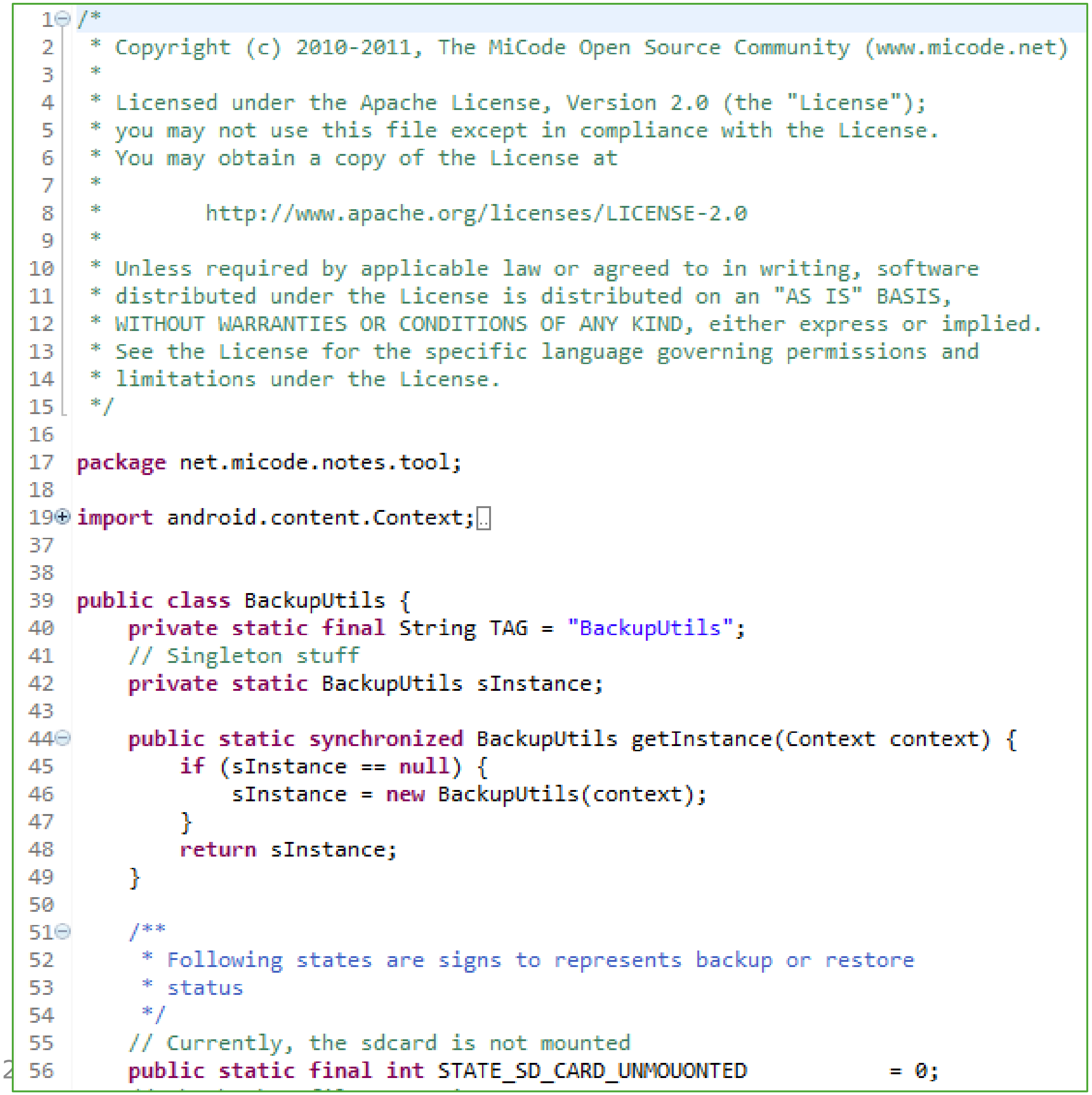
- Class header, function/function header, key statement header, key statement tail.

Effective, necessary, and concise annotations

- Neither too little nor too much is advisable
- Annotations should be understandable, accurate, and unambiguous

Modify with code modifications

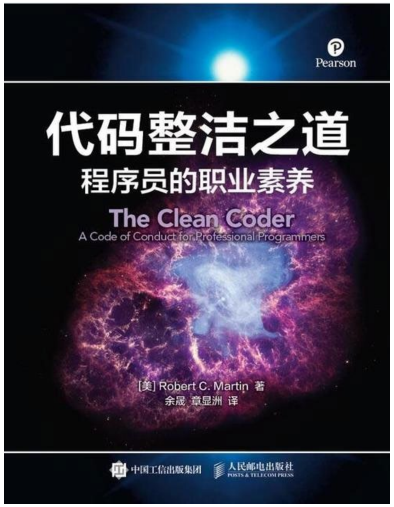
Coding Style - Example

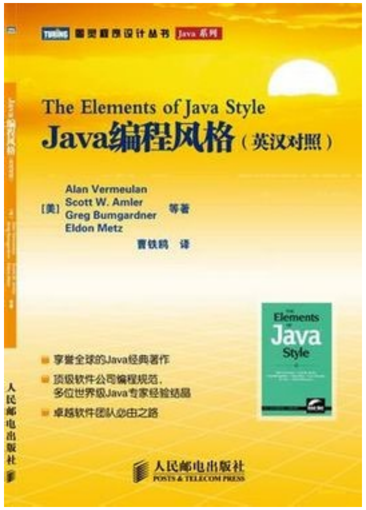


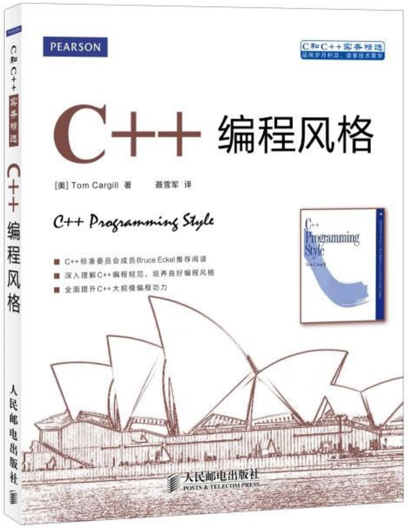
```
/*
 * Copyright (c) 2010-2011, The MiCode Open Source Community (www.micode.net)
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *        http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

package net.micode.notes.tool;

import android.content.Context;


public class BackupUtils {
    private static final String TAG = "BackupUtils";
    // Singleton stuff
    private static BackupUtils sInstance;

    public static synchronized BackupUtils getInstance(Context context) {
        if (sInstance == null) {
            sInstance = new BackupUtils(context);
        }
        return sInstance;
    }

    /**
     * Following states are signs to represents backup or restore
     * status
     */
    // Currently, the sdcard is not mounted
    public static final int STATE_SD_CARD_UNMOUONTED           = 0;
```

- ✓ Notes
- ✓ Name
- ✓ Layout
- ✓ Structure

Books about coding styles







Different programming languages have different coding style requirements

Adopting programming methods

Sentence Design

Modular design

Principle of high cohesion and low coupling

Code Design Specification – Statement(语句) Design

Programming design issues, reflecting the intrinsic quality of the program

- Single entry, single exit.
- Use fewer **goto** statements.
- Strengthen the handling of exceptions.

Analyze and verify the correctness of input parameters, such as name not being empty, age greater than 0 but less than 150.

Execute result processing, set necessary assertions(断言) to analyze.

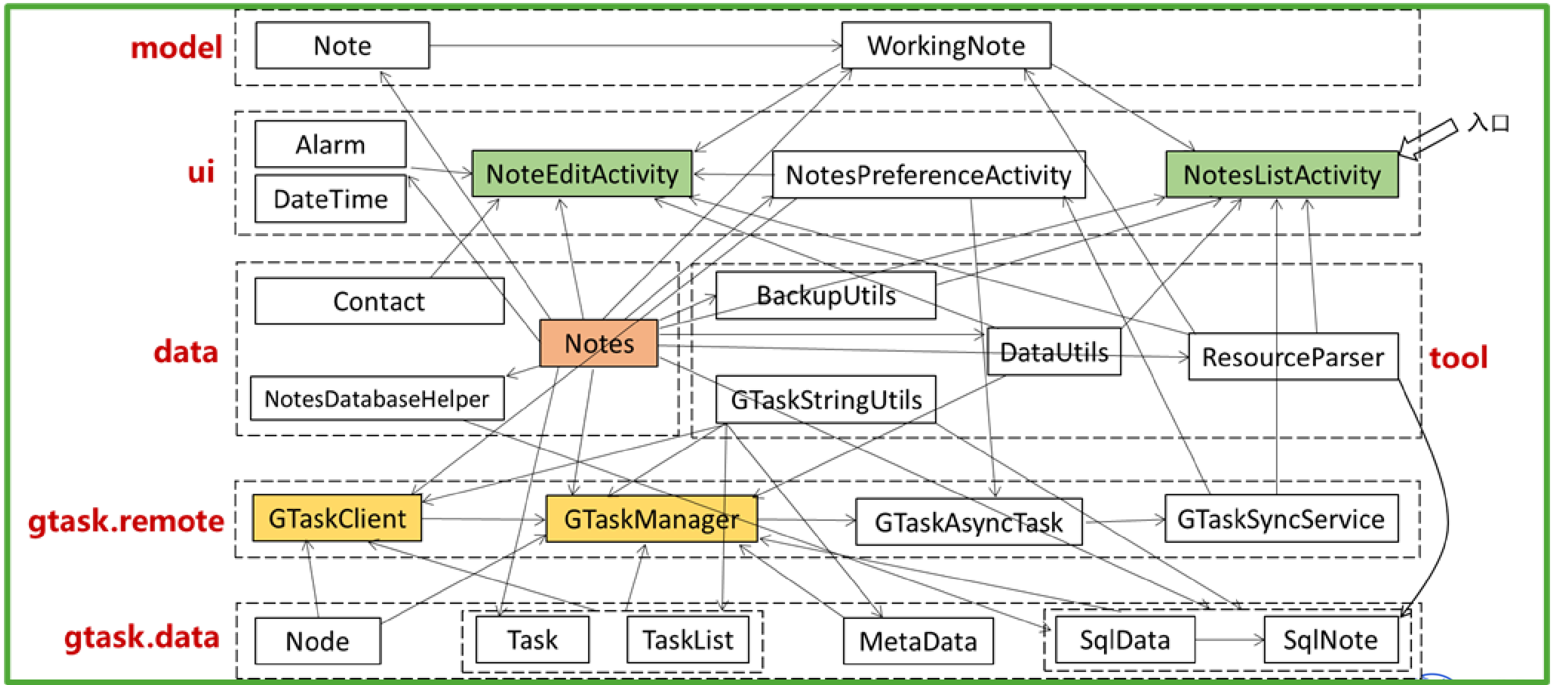
```
p = AllocateNewSpace( );
    Assert(p !=  NULL);
    if (p !=  NULL)  {
                  ……
    }
```

Handle exception statements Try {...} catch (Exception e) {...}

Code Design Specification - Modular design

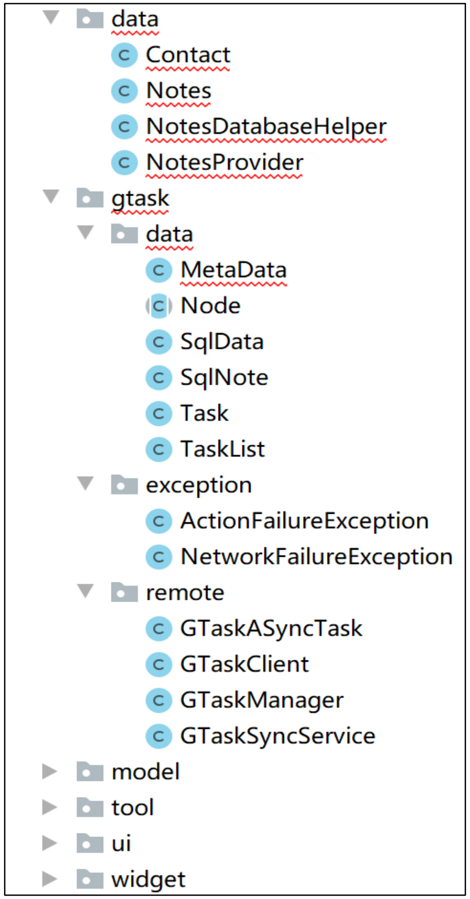
A module is a programming unit that is logically relatively independent and has well-defined interfaces.

Modules can be represented as functions, procedures, methods, classes, packages, etc.

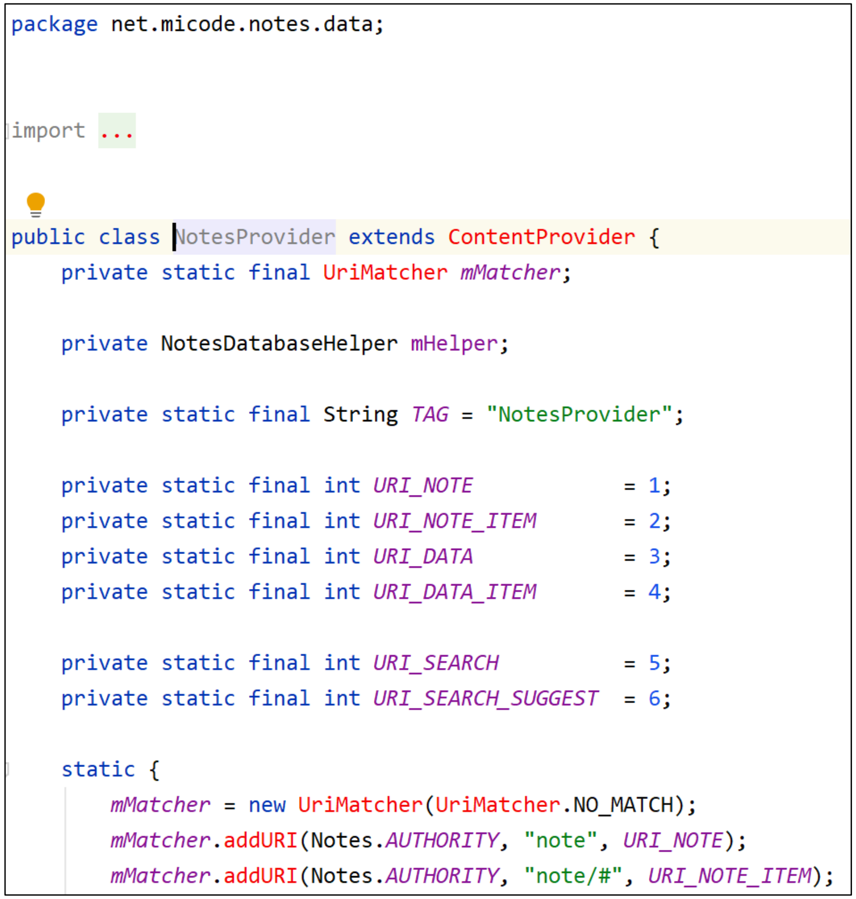


Adopting programming methods - Example

- Sentence Design
- Modularization
- High cohesion
- Low coupling degree



```
▼ data
    Contact
    Notes
    NotesDatabaseHelper
    NotesProvider
▼ gtask
  ▼ data
      MetaData
      Node
      SqlData
      SqlNote
      Task
      TaskList
  ▼ exception
      ActionFailureException
      NetworkFailureException
  ▼ remote
      GTaskASyncTask
      GTaskClient
      GTaskManager
      GTaskSyncService
▶ model
▶ tool
▶ ui
▶ widget
```



```java
package net.micode.notes.data;


import ...


public class NotesProvider extends ContentProvider {
    private static final UriMatcher mMatcher;

    private NotesDatabaseHelper mHelper;

    private static final String TAG = "NotesProvider";

    private static final int URI_NOTE            = 1;
    private static final int URI_NOTE_ITEM       = 2;
    private static final int URI_DATA            = 3;
    private static final int URI_DATA_ITEM       = 4;

    private static final int URI_SEARCH          = 5;
    private static final int URI_SEARCH_SUGGEST  = 6;

    static {
        mMatcher = new UriMatcher(UriMatcher.NO_MATCH);
        mMatcher.addURI(Notes.AUTHORITY, "note", URI_NOTE);
        mMatcher.addURI(Notes.AUTHORITY, "note/#", URI_NOTE_ITEM);
```

Carry out code reuse

What is code reuse?

In the process of writing code, make full use of existing and ready-made code and integrate it into the program to achieve program functionality

What are the benefits of code reuse?

Due to the repeated use of reused code, the quality of the code is fully verified. Therefore, code reuse can not only greatly improve programming efficiency, but also effectively enhance program quality.

Challenges when writing programs

Where does program functionality come from?

Who and how to determine software functionality?

What should be done, if the software scale is large?

- 500 LOC vs 1M LOC

How to rewrite code for software function changes?

Where and how to make changes?

How to ensure program quality?

Can you think of the quality issues with the code and how to ensure it?

Thinking and Discussion

How to clarify functions and divide modules?

How to write code and ensure quality?

Is it feasible for large-scale applications or systems?

What difficulties and problems are you facing?

Conclusion of Coding

Multiple quality requirements for programs

- External and internal, grammar and semantics

Methods to ensure code quality

- Coding standards, design methods, code reuse

Learn to write high-quality program code

- How to write code? How to ensure quality?

What is Software?

Software is

(1) **instructions** (computer programs) that when executed provide desired features, function, and performance;

(2) **data structures** that enable the programs to adequately manipulate information;

(3) **documentation** that describes the operation and use of the programs.

Components of a Computer

Hardware vs **Software**.

A Computer system contains hardware and many types of software. Normally, software is in three types, Application Software, Middleware and System Software.

Question

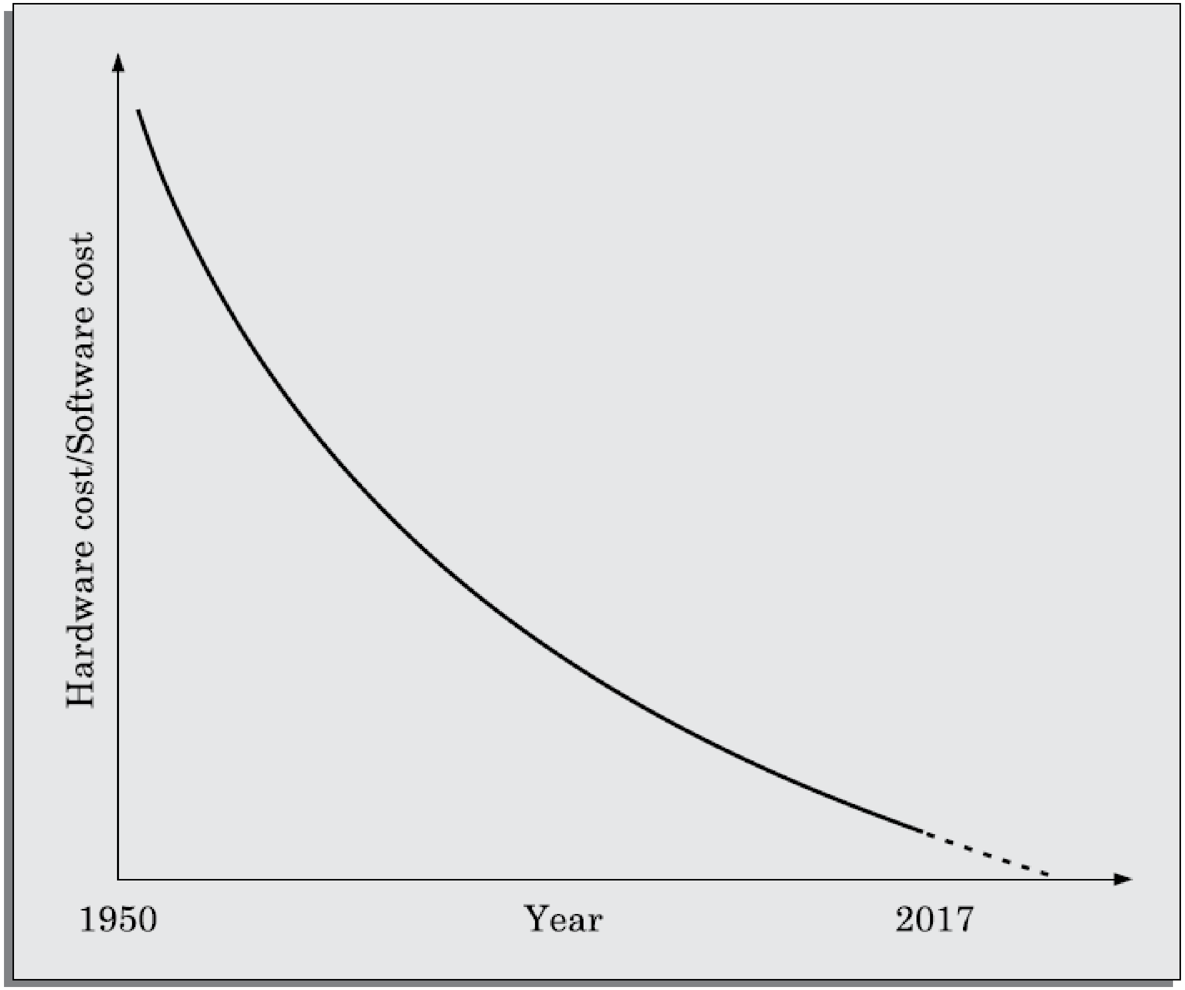
Give some examples of App. Middleware or System?

Almost all software products or application programs running on your computer have been developed by a team.

Example

Microsoft Office Suite.

Software & Hardware Cost comparation



What is Software?

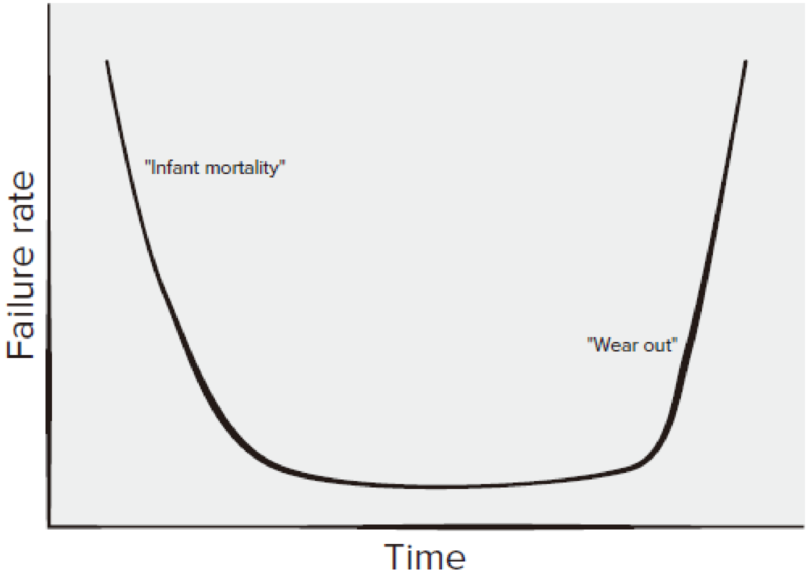
Building a software product is an activity similar to the construction of an apartment. Users must get **functions** !

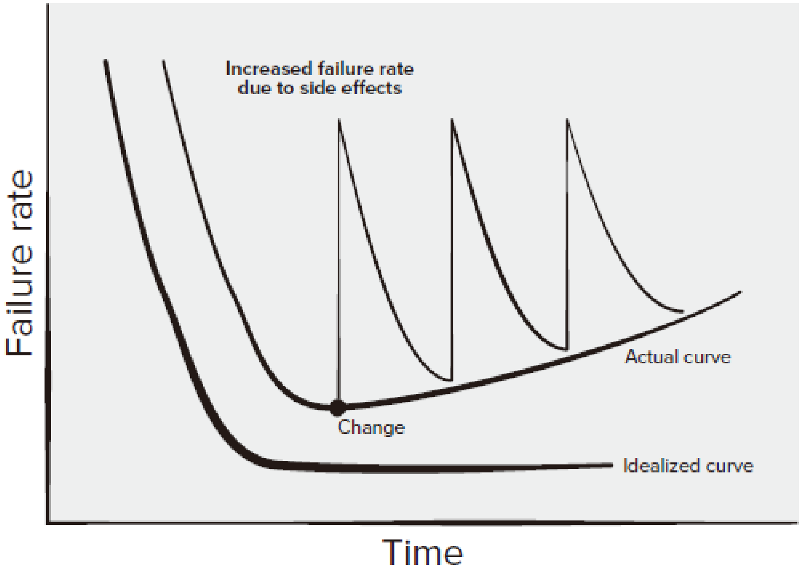
We should also think the **non-functional** issues, similar to the measures taken while constructing the apartment, these considerations including security, performance and reliability.

But, there are **differences** between hardware and software.

Example: Wear vs. Deterioration

Wear vs. Deterioration





What is Software engineering?

- Software is developed or engineered, it is not manufactured in the classical sense.
- Although the industry is moving toward component-based construction, most software continues to be custom-built.

What is Engineering?

Engineering helped create large man-made structures using applied science.

Today, we use devices such as mobile phones, Internet and automobiles, most of these are driven or controlled by computers.

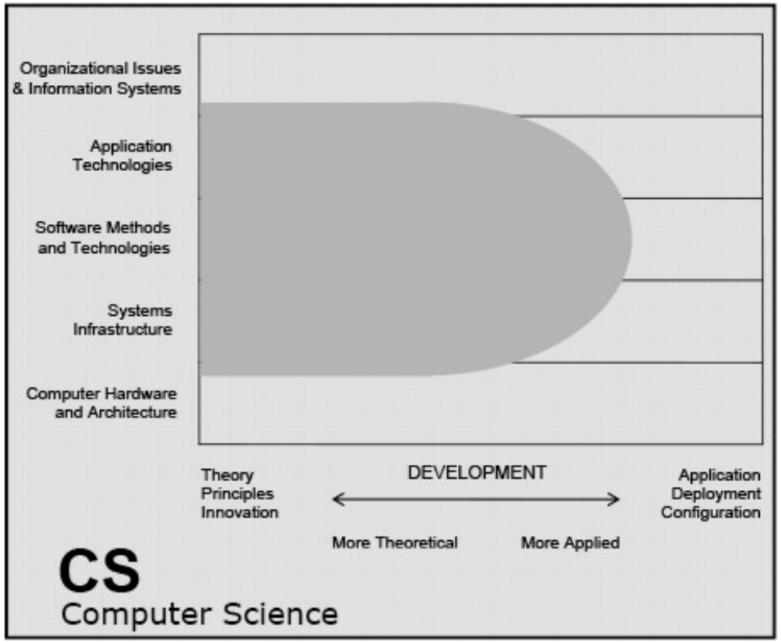
Question

Could you find a device without software in your daily life?

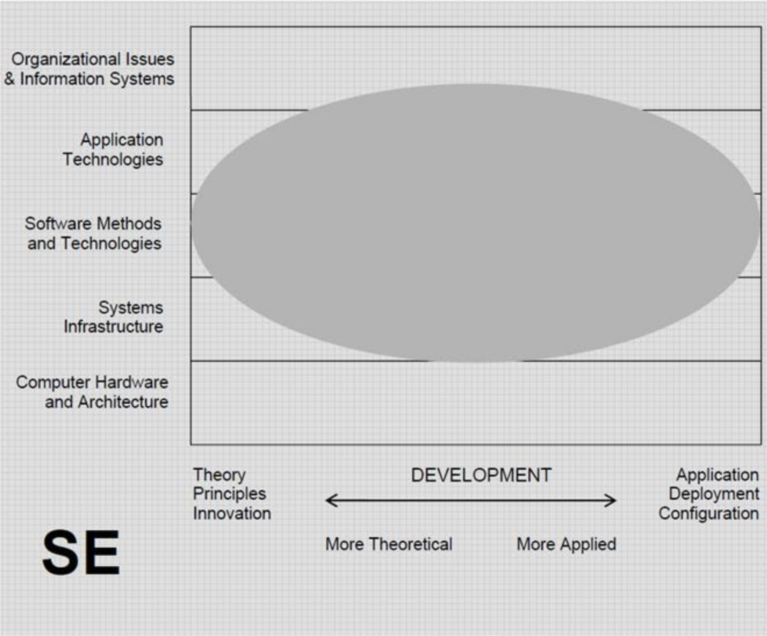
Question

*What is the difference between **science**, **engineering** and **technology**?*

Relationship of SEE and CS?


Organizational Issues & Information Systems
Application Technologies
Software Methods and Technologies
Systems Infrastructure
Computer Hardware and Architecture
Theory Principles Innovation
DEVELOPMENT
Application Deployment Configuration
More Theoretical
More Applied
CS
Computer Science

Software Engineering


Organizational Issues & Information Systems
Application Technologies
Software Methods and Technologies
Systems Infrastructure
Computer Hardware and Architecture
Theory Principles Innovation
DEVELOPMENT
Application Deployment Configuration
More Theoretical
More Applied
SE

- Develop and maintain software systems that
  - –Behave reliably and efficiently
  - –Are affordable to develop and maintain
  - –Satisfy all customer requirements
- Large, expensive, complex systems
- Safety-critical and mission-critical applications

Challenges in software development & evolution?

- Why does it take so long to get software finished?
- Why are development costs so high?
- Why can’t we find all errors before we give the software to our customers?
- Why do we spend so much time and effort maintaining existing programs?
- Why do we continue to have difficulty in measuring progress as software is being developed and maintained?

Challenges in Software Engineering

- Developing a software product is always a challenging job. The reasons are numerous.
- The conversion of user requirements into a software product involves a long chain of processes.
- **E**xample

Suppose a user requires a software that is able to analyze the popularity(人气) of his/her personal web page.

- **Q**uestion

How are you going to build such a software?

Challenges in Software Engineering

- Example (cont.)
  - First, design a program counting the number of hits to that web page within a specified period of time, e.g. 24 hours.
  - Then, report this count to an e-mail address of the user.
  - Third, write the source code using a programming language.
  - Last, after thorough testing, you can deliver the product to the user.

Challenges in Software Engineering

Example (cont.)

- Building such a simple software is easy. However, consider that the user’s requirements are not only to provide the number of hits to the web page, but also to categorize such hits according to the following criteria:
  - Number of hits for the last year.
  - Number of visitors by country per day, per week, and per month.
  - Number of visitors by different browsers used by the visitors.
  - Number of hits and the number of unique visitors.

Challenges in Software Engineering

Example (cont.)

- Now, if you try to design the software according to the new requirements of the user, you will realize that it is not easy to develop the required software this time.

- As the complexity and size of the user requirements increase, the design and implementation sizes of the product also increase. This increase in complexity and size is the main problem associated with the development of software products.
- This is the reality that most of software product development projects face.

Why Software Engineering important?

- Building large software products does require software engineering. Definitely, software engineering techniques make it possible to create useful software products with sufficient quality, within the time limits and within the budget if software engineering practices are adopted.
- Functions of software engineering
  - Reduction of development & maintenance costs.
  - Reduction of development time.
  - Increasing the quality: quality is the single most important ingredient in making any product successful.

The main content of this course

- The course presents a framework that can be used by those who build computer software – people who must get it right.
- The framework encompasses
  - Software process and its models.
  - Requirement engineering.
  - Software design and methods.
  - Software configuration and maintenance.

Software application domains

- Seven broad categories of software
  - System software.
  - Application software.
  - Engineering/Scientific software.
  - Embedded software.
  - Product-line software.
  - Web/Mobile applications.
  - Artificial intelligence software.

Careers in Software Engineering

- Software engineering provides several careers to pursue.
  - A person having skills and interest in finding defects in software, can pursue a career in software testing.
  - Anyone who enjoys programming can become a software developer and write source code.
  - A person good in designing and architecture can become a software designer or architect.
  - A person who likes the task of collection and classification of information can become a business analyst.
  - A person who has gained experience in more than one area of software engineering can become a software project manager.
  - New careers are emerging as software engineering technology emerges.
- **A**ssignment: Look for Software Engineer’s salary.

Legacy Software

Why must legacy software change?

- Software must be adapted to meet the needs of new computing environments or technology.
- Software must be enhanced to implement new business requirements.
- Software must be extended to make it interoperable with other more modern systems or databases.
- Software must be re-architected to make it viable within a network environment.
- **Q**uestion: Give some examples of legacy software.

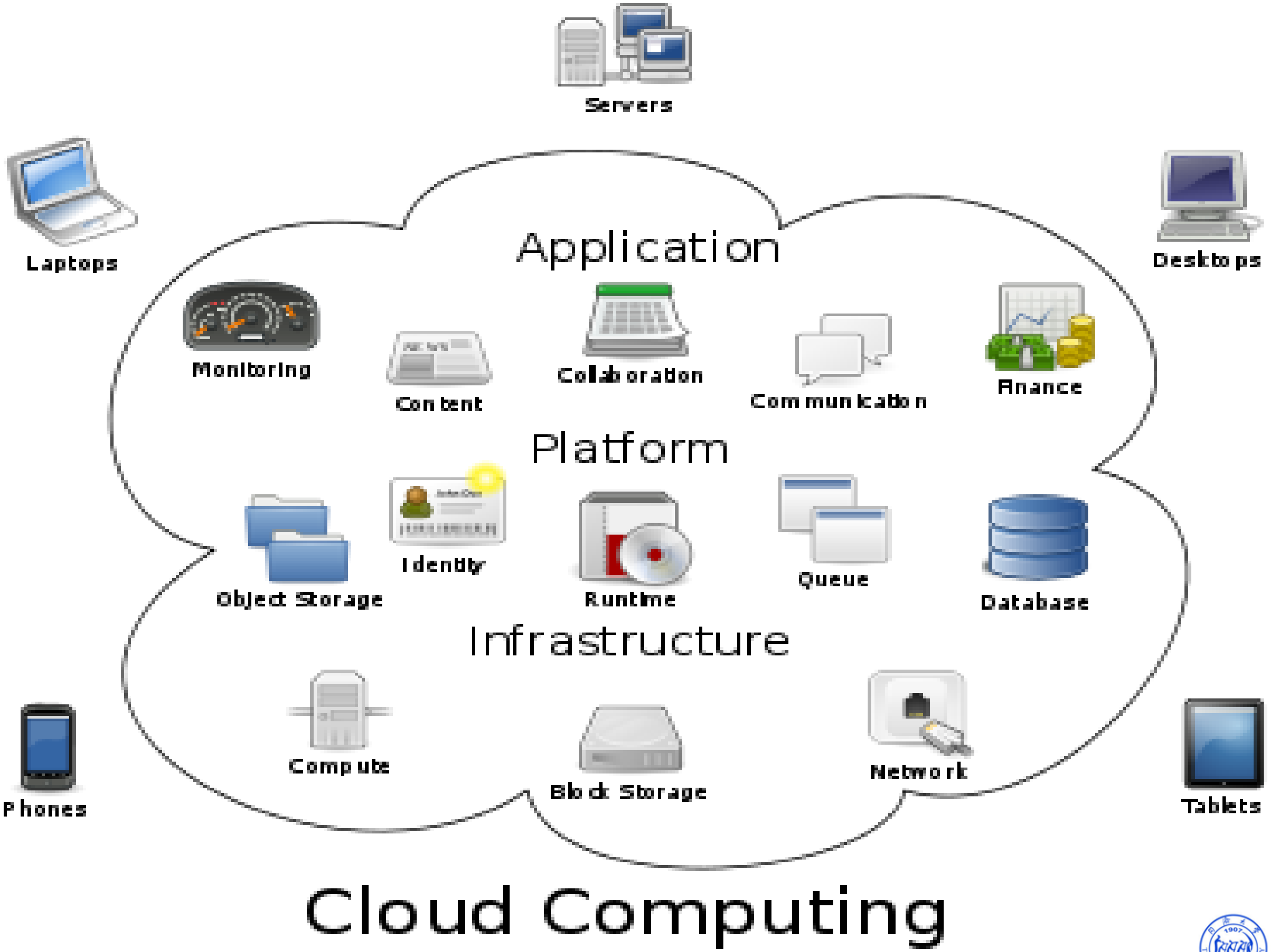
WebApps (8th Edition)

- Modern WebApps are much more than hypertext files with a few pictures.
- WebApps are augmented with tools like XML and Java to allow Web engineers including interactive computing capability.
- WebApps may standalone capability to end users or may be integrated with corporate databases and business applications.
- Semantic web technologies (Web 3.0) have evolved into sophisticated corporate and consumer applications that encompass semantic databases that require web linking, flexible data representation, and application programmer interfaces (API’s) for access.
- The aesthetic nature of the content remains an important determinant of the quality of a WebApp.

Mobile Apps (8th Edition)

- Reside on mobile platforms such as cell phones or tablets.
- Contain user interfaces that take both device characteristics and location attributes.
- Often provide access to a combination of web-based resources and local device processing and storage capabilities.
- Provide persistent storage capabilities within the platform.
- A mobile web application allows a mobile device to access to web-based content using a browser designed to accommodate the strengths and weaknesses of the mobile platform.
- A mobile app can gain direct access to the hardware found on the device to provide local processing and storage capabilities.
- As time passes these differences will become blurred.

Cloud Computing (8th Edition)



Cloud Computing (8th Edition)

- Cloud computing provides distributed data storage and processing resources to networked computing devices.
- Computing resources reside outside the cloud and have access to a variety of resources inside the cloud.
- Cloud computing requires developing an architecture containing both frontend and backend services.
- Frontend services include the client devices and application software to allow access.
- Backend services include servers, data storage, and server-resident applications.
- Cloud architectures can be segmented to restrict access to private data.

Product Line Software (8th Edition)

- Product line software is a set of software-intensive systems that share a common set of features and satisfy the needs of a particular market.
- These software products are developed using the same application and data architectures using a common core of reusable software components.
- A software product line shares a set of assets that include requirements, architecture, design patterns, reusable components, test cases, and other work products.
- A software product line allow in the development of many products that are engineered by capitalizing on the commonality among all products with in the product line.

Assignments

- Chapter 1. Problems to ponder
  - 1.2 Provide examples(both positive and negative) that indicate the impact of software on our society.
  - 1.3 Develop your own answers to the five questions asked at the beginning of Section1.1 in the textbook.

    Discuss with your classmates.
- Questions
  - What is the Definition of Software Engineering proposed by Roger S. Pressman?
  - What is the principal aim of the software engineering discipline? What does the discipline of software engineering discuss?
  - Why do you think systematic software development using the software engineering principle is any different than art or craft?

Assignments

- Team building

Building team for course project development.

Constraints:

1. The number of teammates is not more than 4 students (including 4 students).

2. You can choose teammates from Class 2 and Class 3, but you cannot choose your teammates from Class 1.

3. The team list will be submitted by the teamleader before 24:00 on 21th September by replying to the canvas assignment.

4. The naming rule of this assignment file: "Teamleader student’s ID + His/Her name".

Assignments

- Preview

  《Software Engineering》 8th edition & 9th edition, written by Roger.S.Pressman et.al

  Chapter 1 Software and Software Engineering

  Chapter 2 Process Models

- Reading

  《The Mythical Man-Month Essay on Software Engineering》 written by Frederick P.Brooks

  Chapter 1. The Tar Pit

  Chapter 2. The Mythical Man-Month