CSCI1530 Computer Principles and Java Programming

Tutorial 2
Binary system & Programming style

Zheng Qingqing
SHB 911
qqzheng@cse.cuhk.edu.hk

Content

- Binary Number System
- Programming style
- Generating Javadoc
- Issues of Assignment 1

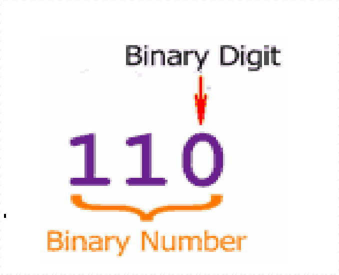
Binary Number System

Binary number system

- A method of representing numbers that has the base of 2 and represents numeric values using two different symbols: typically 0(zero) and 1(one).

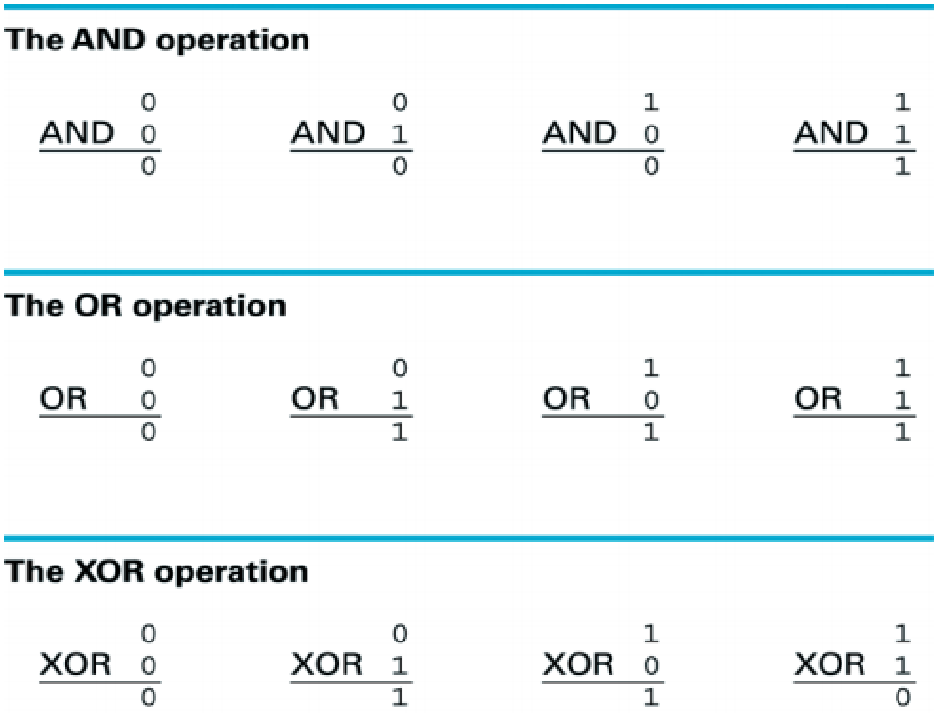
- Current computer systems are designed based on binary number systems.

- In computer world, binary digit is short for bit



Boolean operation:

- Boolean operations AND, OR and XOR(exclusive or)



The AND operation

0	0	1	1
AND 0	AND 1	AND 0	AND 1
0	0	0	1

The OR operation

0	0	1	1
OR 0	OR 1	OR 0	OR 1
0	1	1	1

The XOR operation

0	0	1	1
XOR 0	XOR 1	XOR 0	XOR 1
0	1	1	0

True only when both of its inputs are true

True when at least one of its inputs is true

True when one of its inputs is true and other is false

Representing information as bits

- Representing Text: Using some encoding regulation

-- ASCII 8 bits

--Unicode, 16 bits

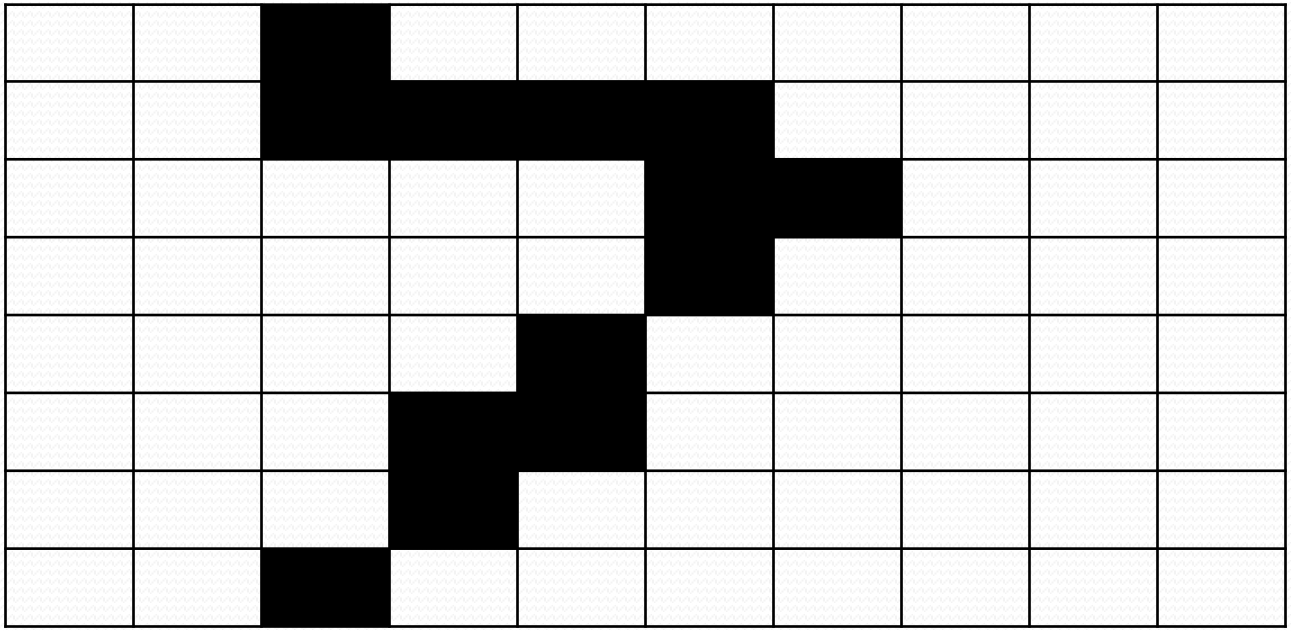
- Representing Numeric Values

--Binary system ↔Ten(Decimal) system

- Representing images: images are made of pixels

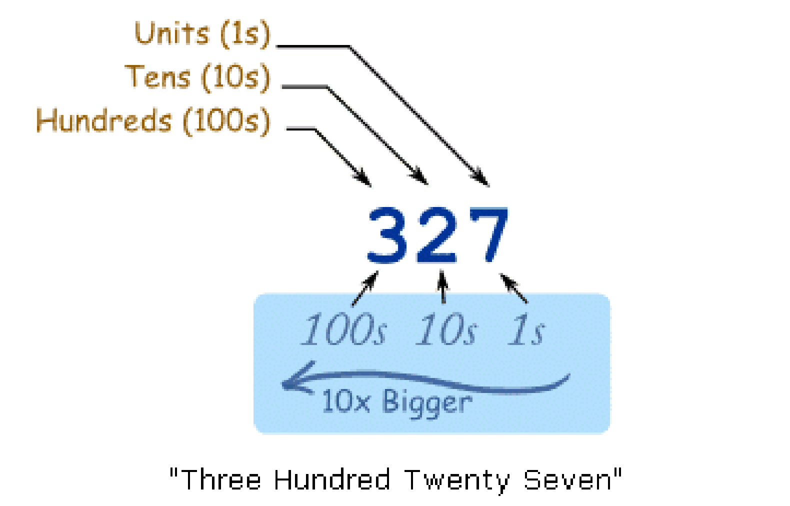
--Bitmap technologies: several bits for every pixel as color

Representing images



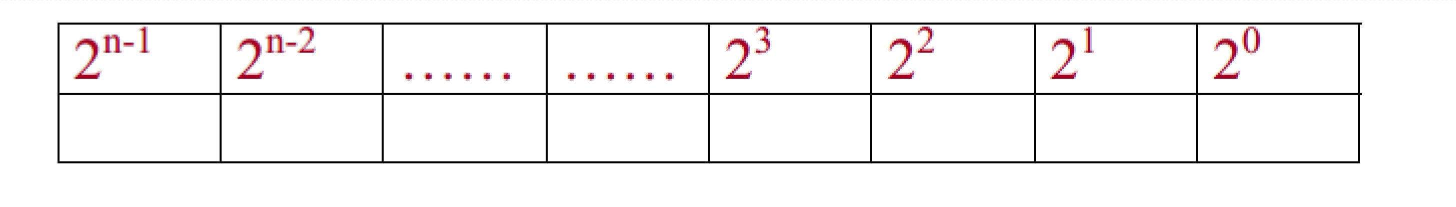
Decimal number system

- Decimal Numbers
  - A method of representing numbers that has the base of 10 and uses ten digits: 0, 1, 2, …, 9.
  - The numerical base most widely used nowadays.



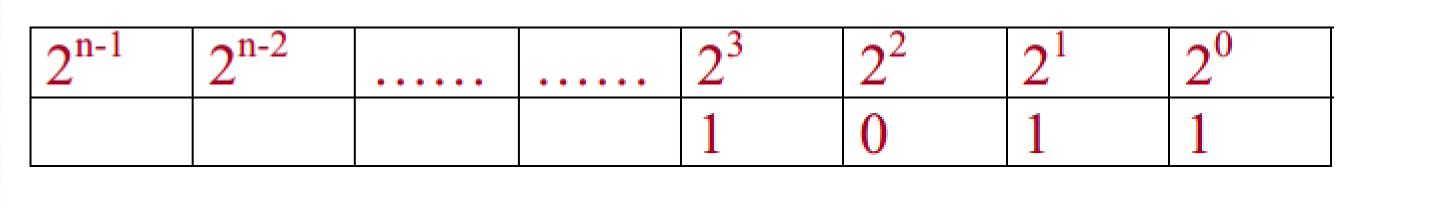
Conversion of binary to decimal

- Replace each position of binary digit by an equivalent power of 2 and then add up.



2^{n-1}	2^{n-2}	……	……	2^3	2^2	2^1	2^0

- Consider a binary number: 1011



2^{n-1}	2^{n-2}	……	……	2^3	2^2	2^1	2^0
				1	0	1	1

$$\begin{aligned}(1011)_2 &= 2^3 \text{ x } 1 + 2^2 \text{ x } 0 + 2^1 \text{ x } 1 + 2^0 \text{ x } 1 \\ &= 8 + 0 + 2 + 1 \\ &= 11\end{aligned}$$

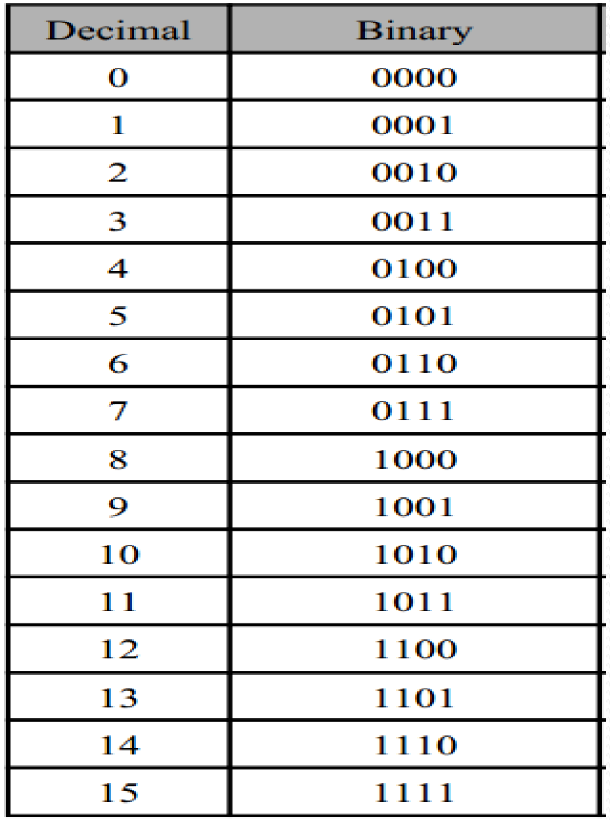
Conversion of decimal to binary

- Keep on dividing the decimal number by 2 repeatedly until it reduces to zero. Then put the remainders in the reverse order.
- Example: convert $(68)_{10}$ to binary
  - 68/2 = 34 remainder is 0
  - 34/2= 17 remainder is 0
  - 17/2 = 8 remainder is 1
  - 8/2 =4 remainder is 0
  - 4/2 = 2 reminder is 0
  - 2/2 = 1 remainder is 0
  - 1/2 = 0 remainder is 1

  Collect the remainders in the reverse order: $(1000100)_2$

Relation Between Binary and Decimal Numbers

- Example



Decimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Links

- An online Converter from Binary /Decimal and Hexadecimal
  - http://www.mathsisfun.com/binary-decimal-hexadecimal-converter.html

- Detailed information for binary, fraction, etc.
  - http://en.wikipedia.org/wiki/Binary_number

Programming Style

Why

- **Programming style** is a set of rules or guidelines used when writing the source code for a computer program.

- It’s not a syntax requirement, but following it will be convenient for programmer’s reading.

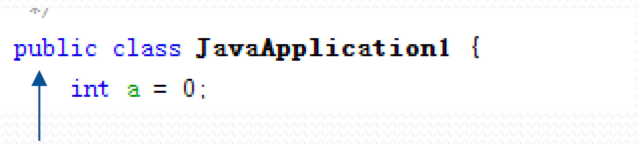
- You can choose what you like, but the following rules are highly recommended.

Naming Conventions

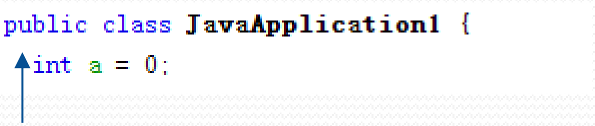
- Use an uppercase letter for the first letter of class names
  - Class `Example`
- Use a lowercase letter for the first letter of variable names.
  - Variable `integer1`
- Use meaningful names (Avoid: `a`, `aa`, `aaa`, `a1`, `a2`, `a3`, `xyz`)
- When a variable consists of multiple words, the first letter of every word, except the first word, should be capitalized.
  - Variable `myIntVar`, but not `myintvar`, not `my_int_var`

Naming Conventions

- Indentation : Insert some spaces in front of the statements when these codes are in a block.
  - Normally 2 or 4 spaces are the more common. Either is ok.

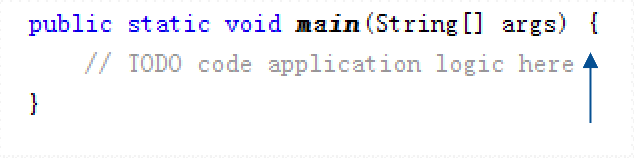


```
 */
public class JavaApplication1 {
    int a = 0;
```

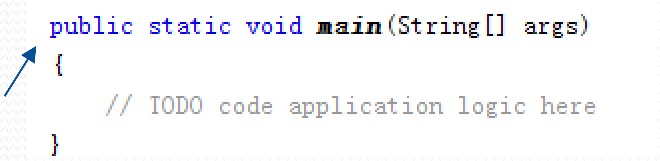


```
public class JavaApplication1 {
  int a = 0;
```

- Bracket Placement : Two ways to place the couples of brackets. Either of them is ok.



```
public static void main(String[] args) {
    // TODO code application logic here
}
```



```
public static void main(String[] args)
{
    // TODO code application logic here
}
```

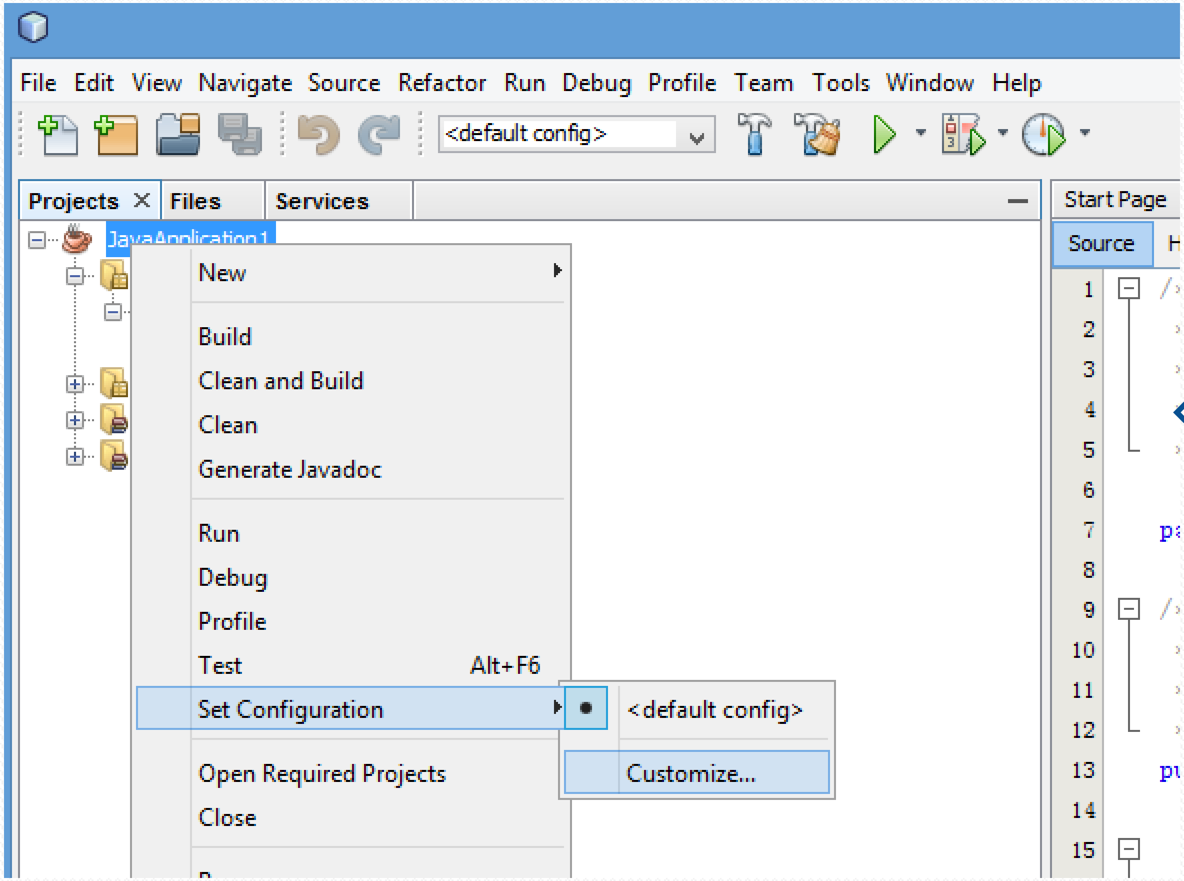
Links

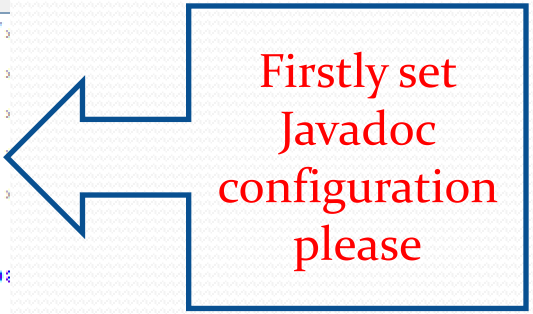
- If you want to know more, please search ‘Java Programming Style’ for details or see the following link.
- http://geosoft.no/development/javastyle.html

Don’t dig too deep about its details.

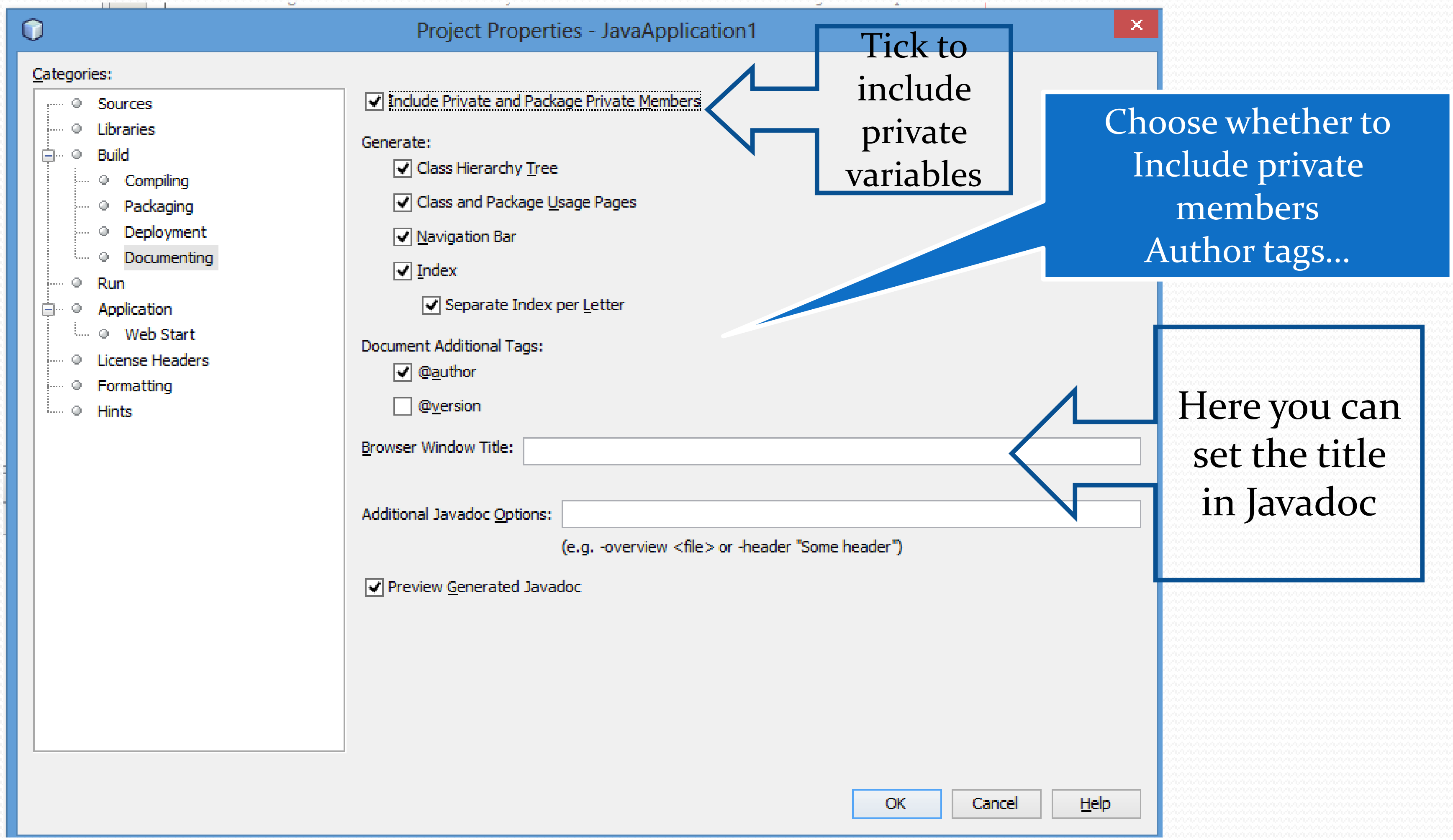
Generating Javadoc

Javadoc configuration




Firstly set Javadoc configuration please

Javadoc configuration



Project Properties - JavaApplication1

Categories:

- Sources
- Libraries
- Build
  - Compiling
  - Packaging
  - Deployment
  - Documenting
- Run
- Application
  - Web Start
- License Headers
- Formatting
- Hints

☑ Include Private and Package Private Members

Generate:

☑ Class Hierarchy Tree

☑ Class and Package Usage Pages

☑ Navigation Bar

☑ Index

☑ Separate Index per Letter

Document Additional Tags:

☑ @author

☐ @version

Browser Window Title:

Additional Javadoc Options:

(e.g. -overview <file> or -header "Some header")

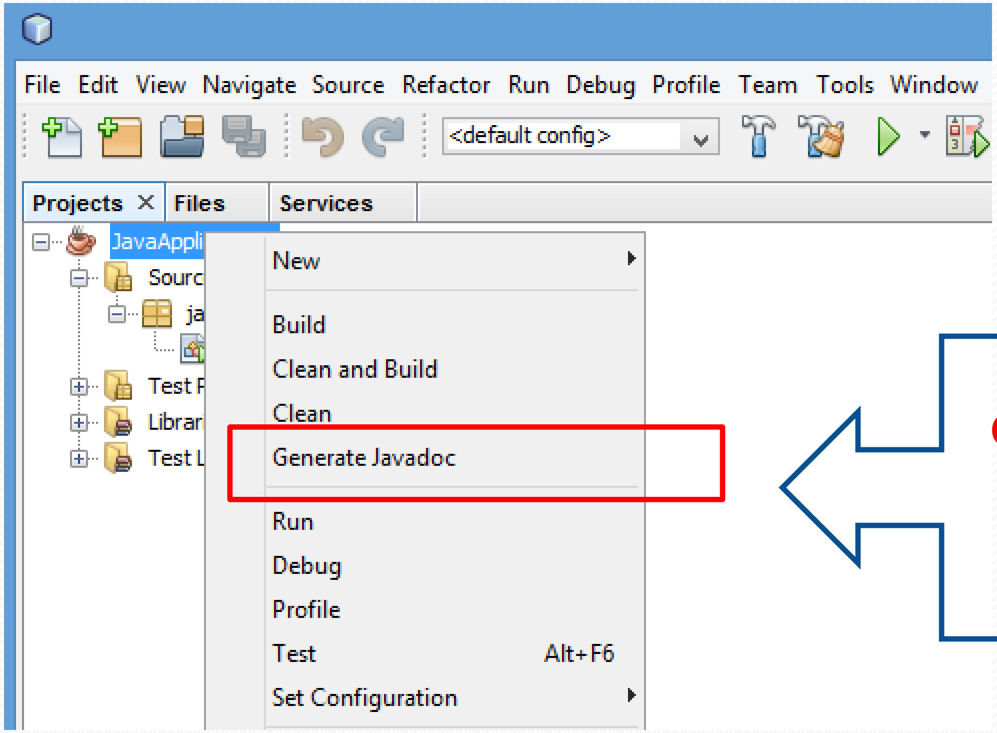
☑ Preview Generated Javadoc

Tick to include private variables

Choose whether to Include private members Author tags...

Here you can set the title in Javadoc

Use NetBeans to generate Javadoc



Click it to generate Javadoc.

Comments for Javadoc

```
/**
 *
 * @author You
 */
```

If you use
// something
Or
/*
something
*/
Those comments will not be included in generated Javadoc .
If you want to include your comments in Javadoc, you need to use the format left

Issues of Assignment 1

Reminder

In this assignment, you don’t need to fully understand the meaning of the example codes.

The main target is to make you familiar with the Java environment and know how to successfully create and run a project .

You will understand them and use those methods to build your own programs when you complete this course.

Procedure

You can follow the instructions step by step.

- Create your projects.
- Complete codes and comment according to pdf

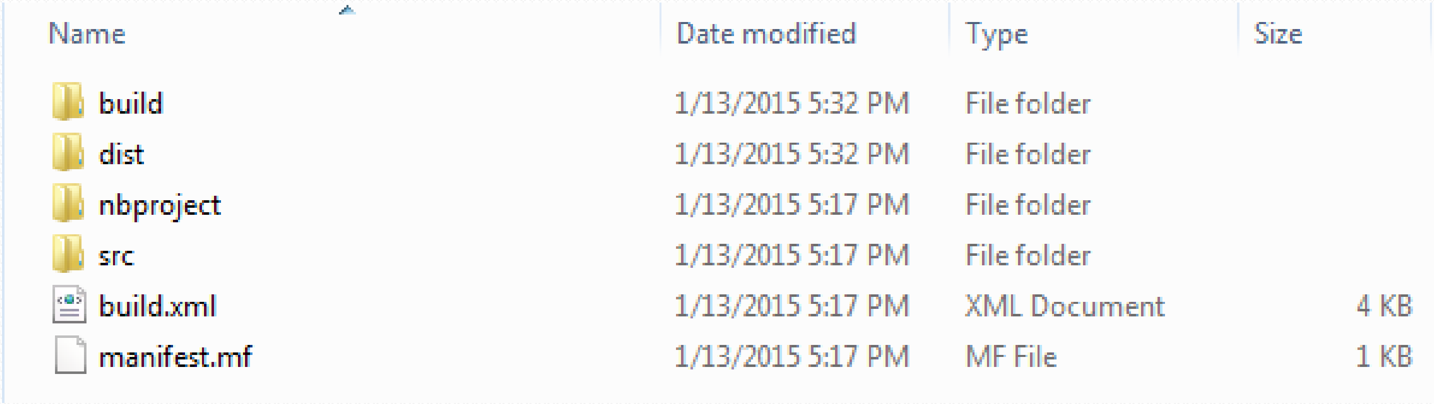
  (Don’t forget to fill in your id & name)
- Run and debug it
- Package and submit the whole project before the deadline

Attention

- 1. The submitted program should be free of any typing mistakes, compilation errors and warnings.
- 2. Comment/remark, indentation, style are under assessment in every programming assignments unless specified otherwise. This program gives you an example of a well-formatted source file. Variable naming, proper indentation for code blocks and adequate comments are important.
- 3. Remember to do your submission before 6:00 p.m. of the due date. No late submission would be accepted.
- 4. If you submit multiple times, **ONLY** the content and time-stamp of the **latest** one would be counted. You may delete (i.e. take back) your attached file and re-submit. We ONLY take into account the last submission.

NetBeans project structure

In the folder of your project



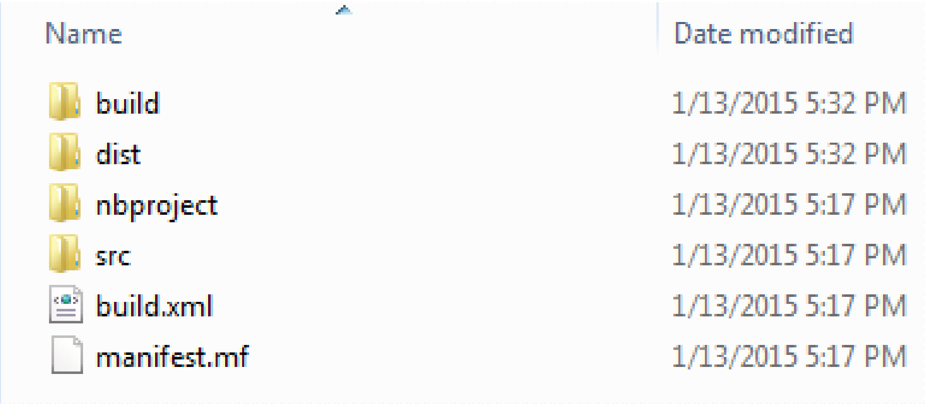
Name	Date modified	Type	Size
build	1/13/2015 5:32 PM	File folder	
dist	1/13/2015 5:32 PM	File folder	
nbproject	1/13/2015 5:17 PM	File folder	
src	1/13/2015 5:17 PM	File folder	
build.xml	1/13/2015 5:17 PM	XML Document	4 KB
manifest.mf	1/13/2015 5:17 PM	MF File	1 KB

build -- .class files after compiling your source codes.

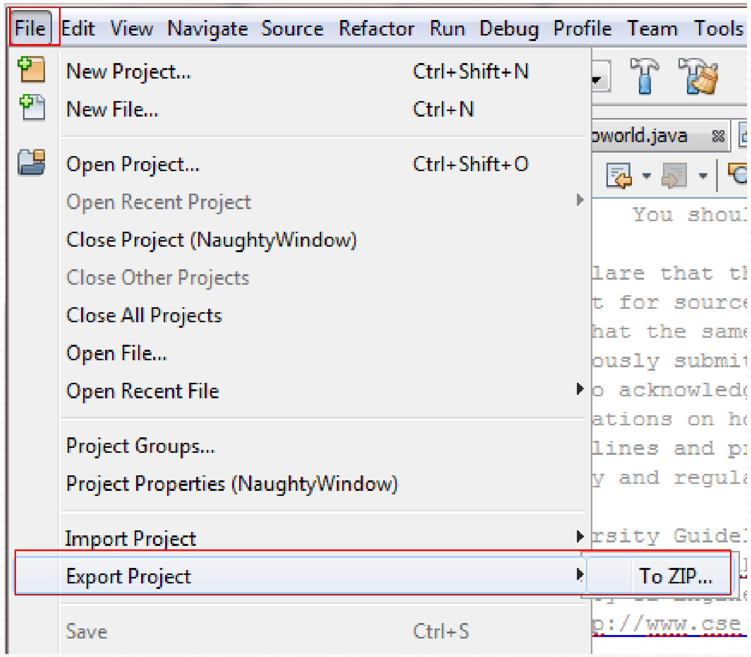
src -- your source code

other file is related to NetBeans

Submit the project



Name	Date modified
build	1/13/2015 5:32 PM
dist	1/13/2015 5:32 PM
nbproject	1/13/2015 5:17 PM
src	1/13/2015 5:17 PM
build.xml	1/13/2015 5:17 PM
manifest.mf	1/13/2015 5:17 PM



Please zip **all** those files and document folders together.

Or use NetBean [File – Export Project – To Zip]

Summary

Summary

- Binary number system is used in computers, which could representing information in bits.
- It's important to follow some common rules in programming style, including how to name variables, the indentation, the bracket placement. Choose your style from them and stick to it.
- Javadoc could be automatically generated and we could use this doc to illustrate the structure of our program.

The end

Thank you!