Dr Heng Aik Koan
akheng@uow.edu.au

Course account:
https://moodle.uowplatform.edu.au

Remember to set Moodle to Singapore time zone

Reference books

Recommended Textbook

Java how to Program (Early Objects), Eleventh Edition, Paul Deitel and Harvey Deitel, Pearson, 2018

Important Reference Books

Introduction to Java Programming Comprehensive, Eleventh Edition, Liang, Pearson 2018

Java programming, Ninth Edition, Joyce Farrell, Cengage learning 2018

Java programming: From problem solving to design, Fifth edition, D.S Malik, Cengage learning 2012

Lecture Plan

1. What you need to know about lectures, labs, assignments and exams
2. What *Programming Fundamentals* is and how it is related to your degree

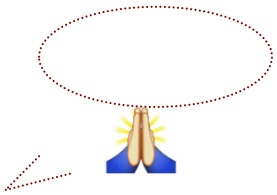
Subject structure

- **Assignments** : 3 assignments
- **Quizzes:** at least 3
- **Labs:** at least 5
- **Term Test**
- **Exam**

This is a 6 credit point subject. According to Course Rule 003 the amount of time spent on this subject should be **at least 12 hours per week** including **self-directed study**

Assessment

Assessment Items	%	Remarks
Assignment 1 (part 1 and 2)	8%	Sequential designs
Assignment 2 (part 1 and 2)	10%	Classes and objects
Assignment 3 (part 1 and 2)	10%	Arrays
Labs and Quizzes	10%	
Term test	12%	
Exam	50%	



Don’t wish to get a lot of marks in the final exam to compensate what have been lost in labs and assignments

Lectures

- The lectures will introduce fundamental computing concepts and the principles of Java programming
- The lectures will contain a sufficient number of examples and slides with animation to facilitate explanation of complex technical aspects
- It is highly recommended that you implement all examples, compile and run the programs on your computer
- You are encouraged to participate actively in the lecture sessions answering questions and making your own notes that will help you to understand material better

(see a list of scheduled lecture topics in the subject outline)

Labs

- All labs are take home lab. You should try your best to finish your lab in one or two days.

- All labs and the assignments need to be demonstrated in class.

Assignments

- There will be three programming assignments
- When an assignment is released, download the assignment description from the subject web site. Read carefully the specifications. Make sure you understand the requirements.
- Take home assignments for programming parts
- Your solutions must be submitted electronically via the *Moodle* system. **No submission via email will be accepted**
- Late assignments **will not be accepted** without a granted special consideration
- Exact time after which the submitted assignment will not be accepted by the system will be indicated in every assignment

Assignments

- When you submit an assignment, you need to follow the submission instructions for this assignment and do not make any assumptions
- In every submitted assignment file you submit must include the following information header :

```
/*-----------------------------------------------
My name:
My student number:
Tutorial group:
Declaration: Make a declaration telling me if this is
             your own work
-----------------------------------------------*/
```

- No header, or an empty header is considered as an anonymous submission. Such assignments are not marked (your mark will be 0)
- Make sure the submitted files are named as required. Files submitted with incorrect names are not recognized by the assignment test system and are not marked (your mark will be 0)

Assignments FAQ

1. How to submit my assignment electronically?
- Via Moodle system

2. What development environment and Java version can I use?

You can use any development environment to work on assignments at home.

Assignments

All assignments must be completed individually

When you submit an assessment task, you are declaring the following:

1. It is your own work and you have not copied anything from others and you have not discussed your work with others
2. You have not plagiarized from published work (**including various internet sources**)
3. You have read your responsibilities under the UOW's policy on plagiarism and you understand possible consequences
4. You have not used storage devices which can be accessed by others without passwords

Plagiarism = Big problems

You may be asked to have a formal meeting with the lecturer to explain your assignment solution if there are doubts that you worked on your assignment yourself

Subject web site

The subject web site is the subject's "Notice Board"

All important notices related to CSIT111 will be posted there. Check it frequently !

Note: Any information posted to the subject web site is deemed to have been notified to all students

Self-directed Study

Listening passively is useless

- Download lecture notes from the subject web site and look through the lecture notes prior to lectures
- Attend all lectures. Take your own notes and add your own comments or questions during the lectures
- Read related chapters in the textbook together with lecture notes and implement Java examples discussed at lectures on your computer at home
- Should you have any questions, ask!!

Subject Materials

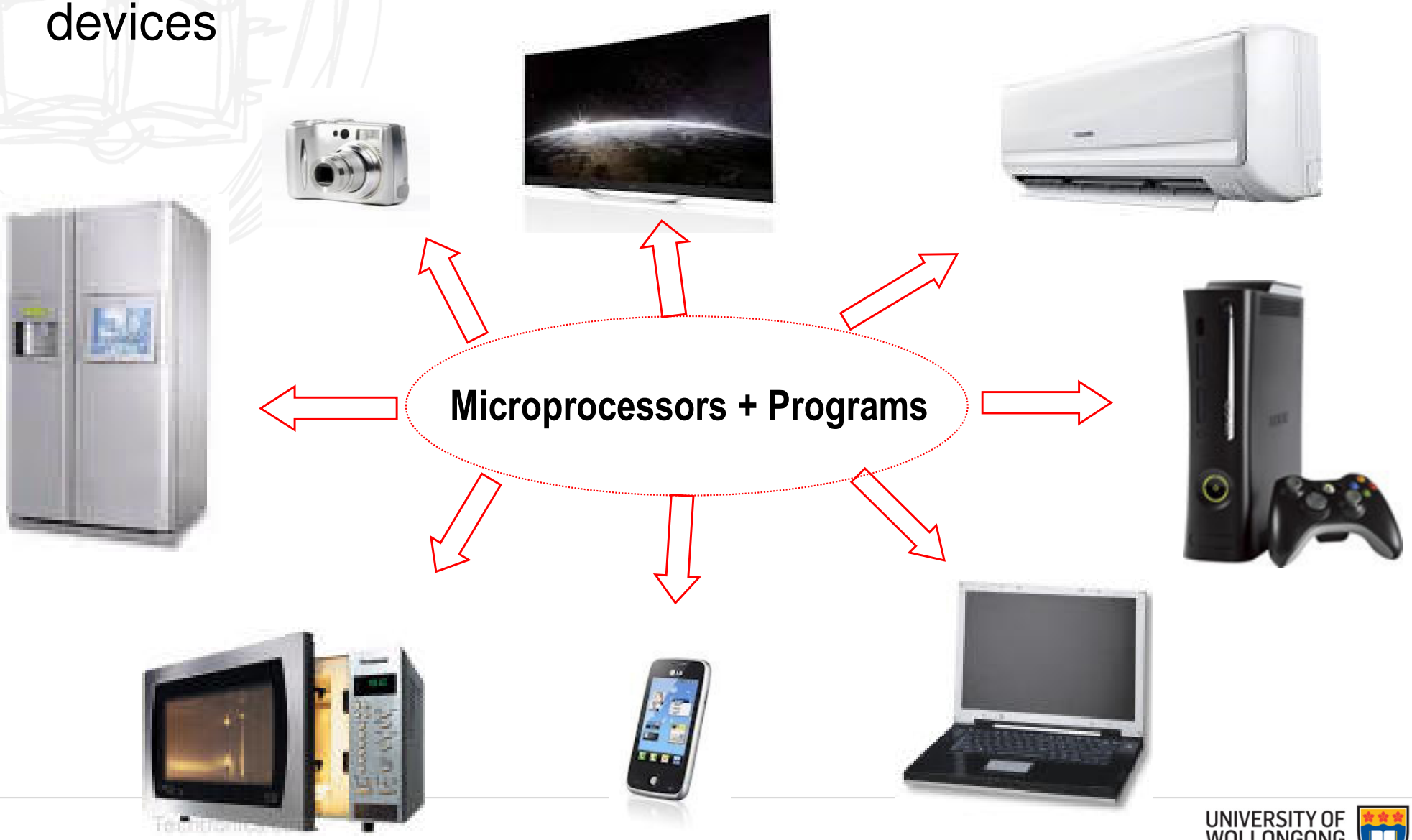
Lecture notes:

- The lecture notes are available on the subject web site
- The lecture notes may not include some examples and explanations given in lectures

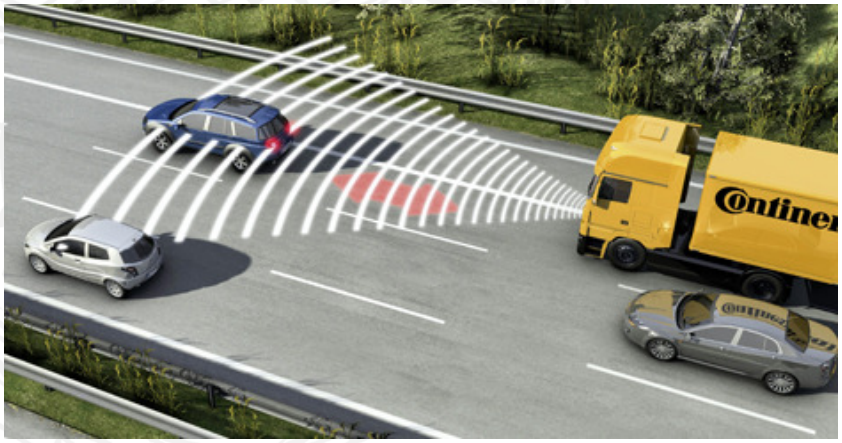
Introduction to Programming Fundamentals

Digital World

In modern life, we are surrounded by digital electronic devices

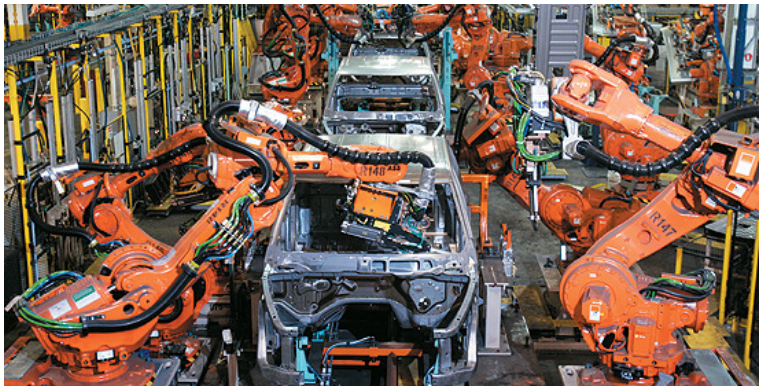


Digital World



Automatic parking
Smart cruise control

Self driving cars



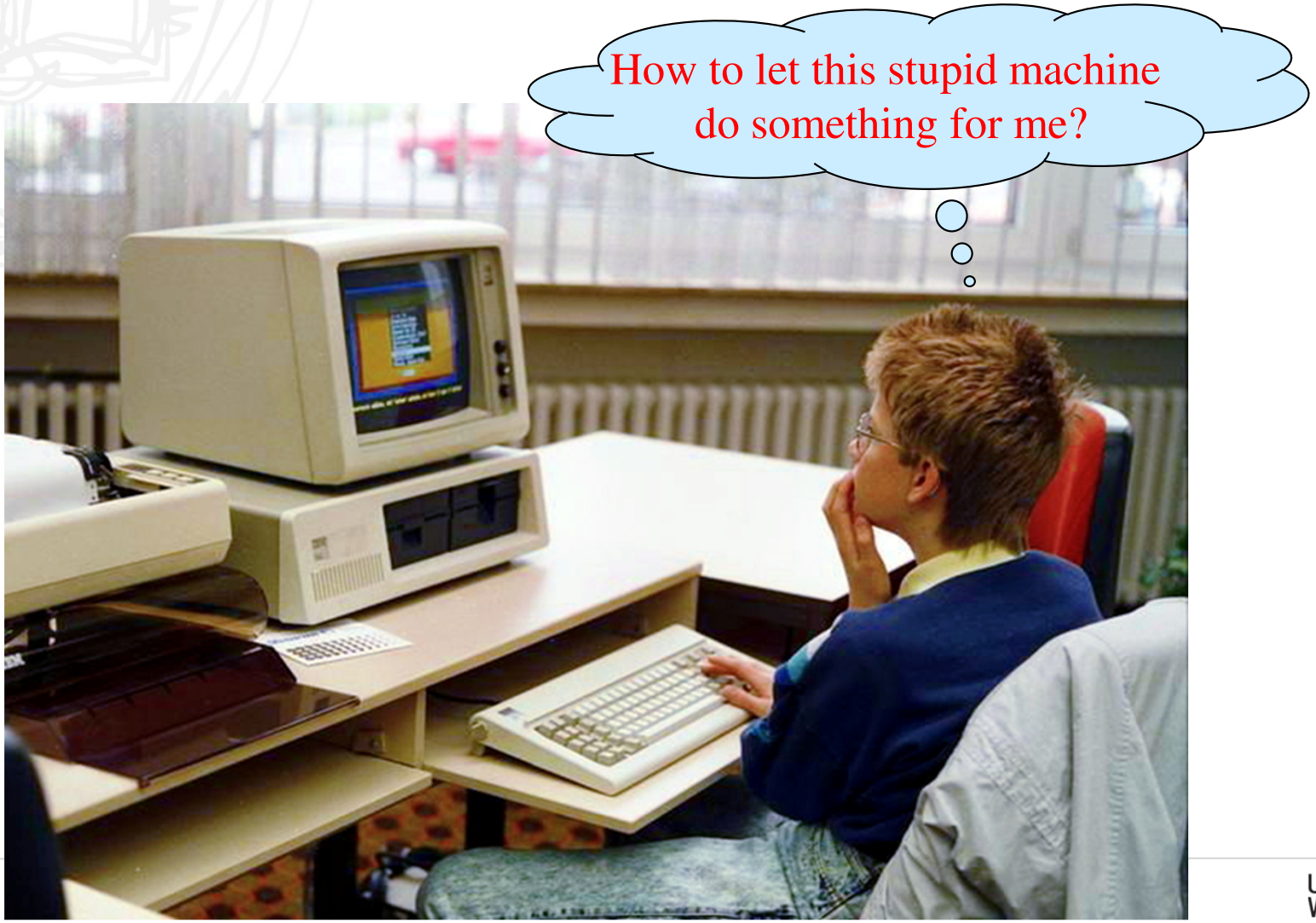
Automated assembly lines

Home robots

Military robots

Computers are incredibly *fast*, *accurate* and *stupid*.
Human beings are incredibly *slow*, *inaccurate* and *brilliant*.
Together they are powerful beyond imagination.

Albert Einstein?… Leo Cherne?... Stuart Walesh?



Personal Computer

1) Scanner
2) CPU (Microprocessor)
3) Memory (RAM)
4) Expansion cards (graphics cards, etc.)
5) Power supply
6) Optical disc drive
7) Storage (Hard disk or SSD)
8) Motherboard
9) Speakers
10) Monitor
11) System software
12) Application software
13) Keyboard
14) Mouse
15) External hard disk
16) Printer

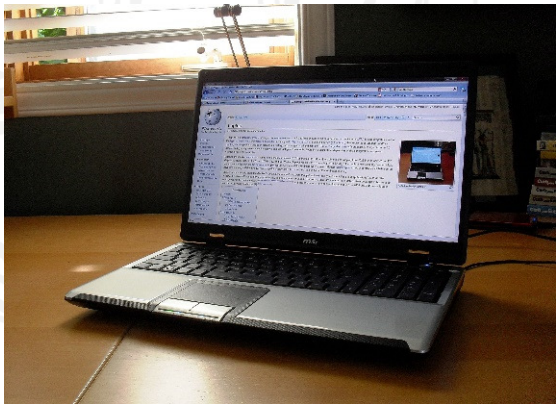
One of the most important components is missing here

Let a computer do something!

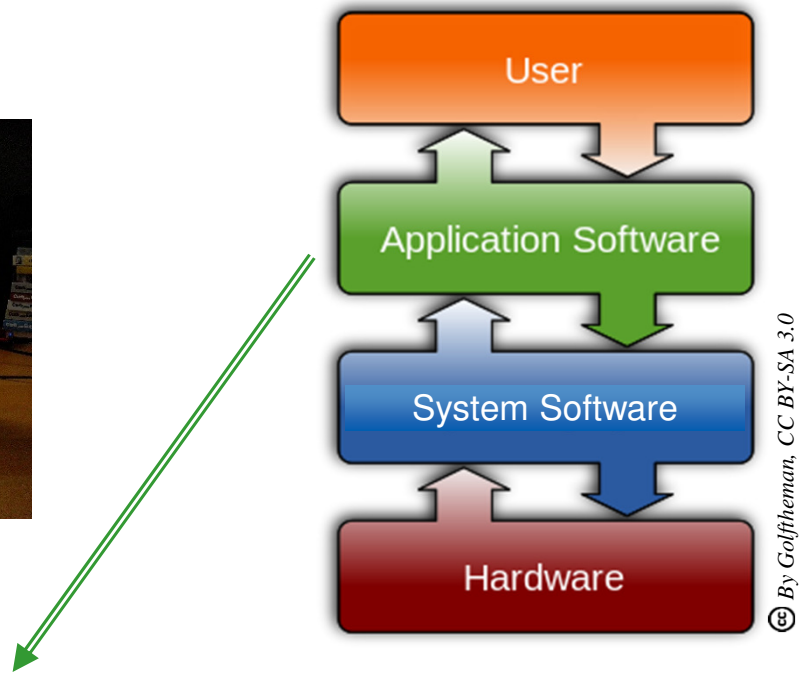
Computers don't do anything without someone telling them what to do

- How to let a computer to do something?
  - Instruct it
- How to instruct a computer to do something?
  - Use a language to instruct it
- How to use a language to instruct a computer to do something?
  - Write a sequence of instructions in a language – a program
- How to write a program?
  - Learn programming – **this subject !**

How computers work



By Kristoferb, CC BY-SA 3.0

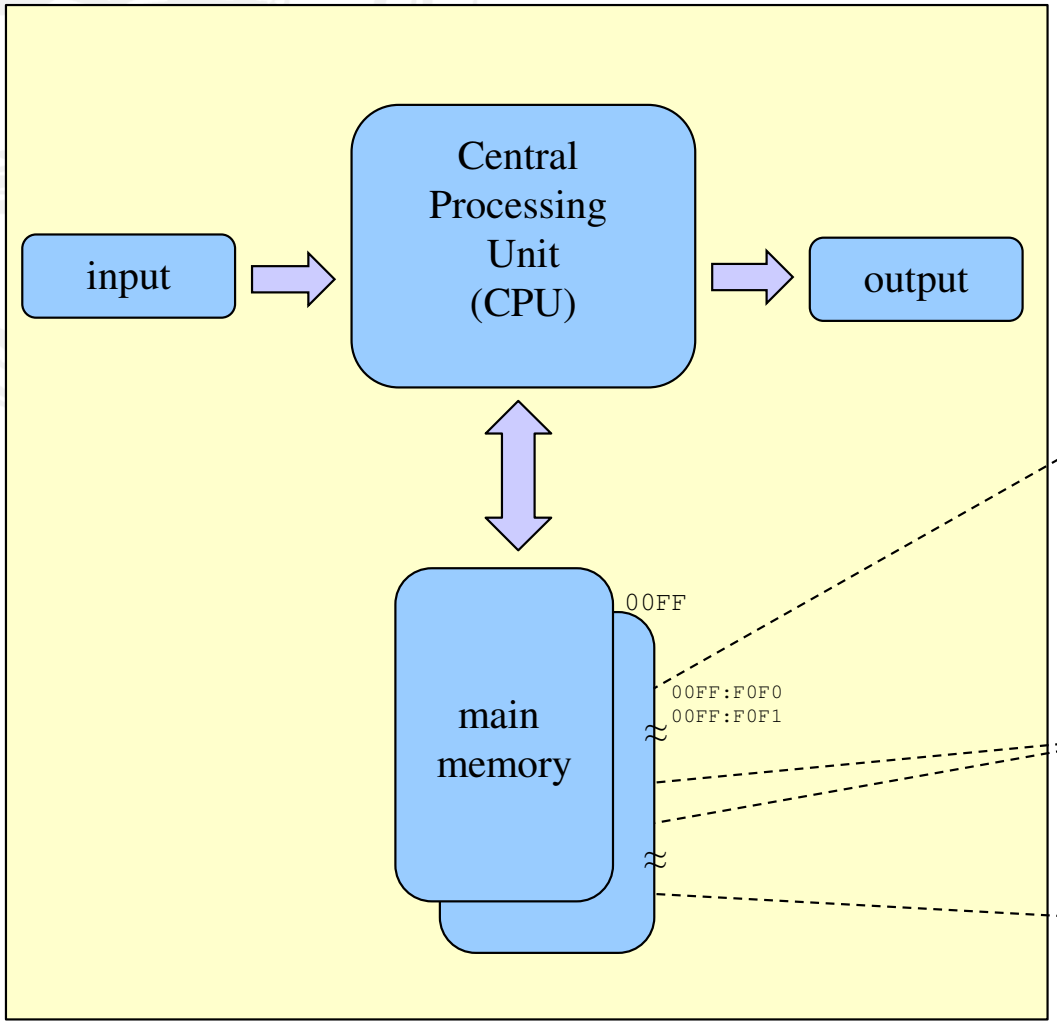


By Golftheman, CC BY-SA 3.0

The focus of CSIT111

```
/**
 * The HelloWorldApp class implements an
 *   application that displays "Hello world!"
 *   to the standard output.
 */
class HelloWorldApp {
  public static void main(String[] args){
          // Display "Hello world!"
          System.out.println("Hello world!");
    }
}
```

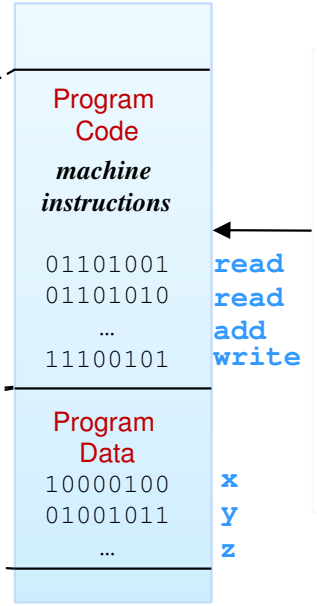
How computer programs work



x86 instructions

ADD	add
SUB	substract
MUL	multiply
CMP	compare
INC	increment
MOV	move data
…	

program in memory



Task:

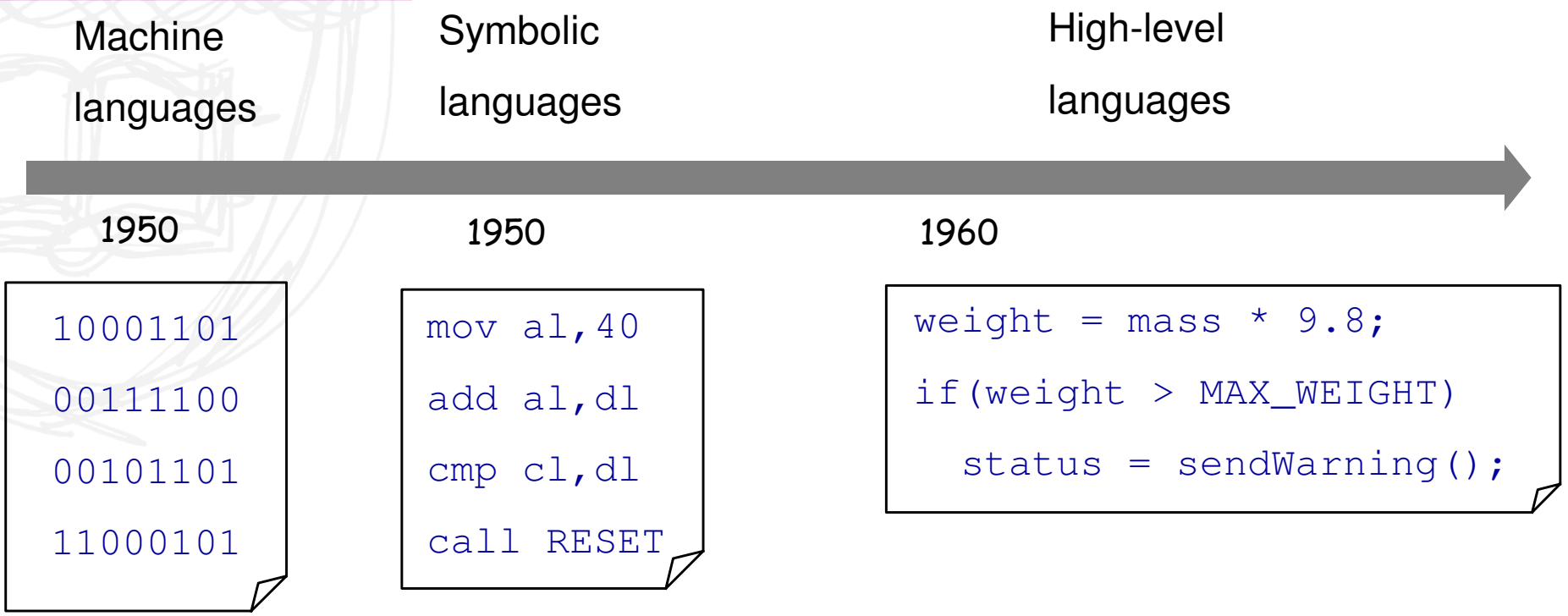
$$z = x + y$$

CPU instructions

```
Read location x
Read location y
Add
Write to location z
```

The instruction binaries are not real and for illustration only

Evolution of Programming Languages



- 1989 – C was standardised by the ANSI/ISO
- 1998 – C++ was standardised by the ANSI/ISO
- 1995 – Java 1.0 was released by Sun Microsystems
  Java has not been formally standardised by the ISO. It is a *de facto* industrial standard controlled through the JCP since 1998

Task: “Hello, world!”

- How do people say hello to the world?

  Hello
  G’day
  你好
  Bonjour
  مرحبا
  こんにちは
  여보세요

- How to instruct your computer to say hello to the world ?

“Hello world” programs

```
.model small
.stack 100h

.data
msg db 'Hello world!$'

.code
start:
    mov ah, 09h
    lea dx, msg
    int 21h
    mov ax, 4C00h
    int 21h
end start
```

Intel x86 instruction set for 16-bit DOS

```
section .data
str:     db 'Hello world!', 0Ah
str_len: equ $ - str
section .text
global _start
_start:
 mov eax, 4
 mov ebx, 1
 mov ecx, str
 mov edx, str_len
 int 80h
 mov eax, 1
 mov ebx, 0
 int 80h
```

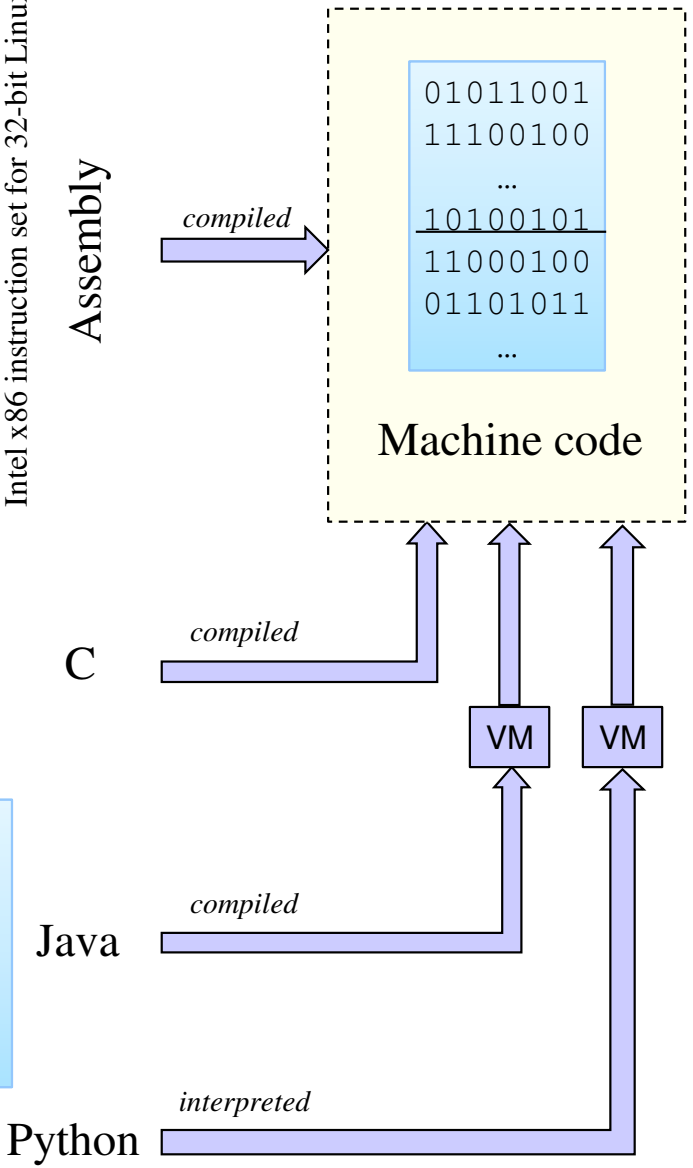
Intel x86 instruction set for 32-bit Linux

Levels of programming languages: *Low* → *High*

```
#include <stdio.h>
int main() {
    printf("Hello World!");
}
```

```
public class HelloWorldApp {
    public static void main(String[] args){
        System.out.println("Hello, World!");
    }
}
```

```
print "Hello World!"
```



Programming languages

Anything that can be done using one language can be done using any language.
Some language may be easier for certain things

- Compiled languages
  - Compiled to machine code
    - Architecture-dependant, high performance
    - **`Assembly, C, C++`**
  - Compiled to bytecode
    - Architecture-neutral (running in a virtual machine)
    - **`Java`**
- Interpreted (scripting) languages
  - Programming languages without explicit compilation, interpreted at run-time
    - **`JavaScript, PHP, Perl, shell`**
    - **`Python`** is used without compilation, but is compiled to bytecode on-the-fly and running in a virtual machine
  - Simple and less lines of code, less access to computer native resources, slower execution

The differences are becoming fewer

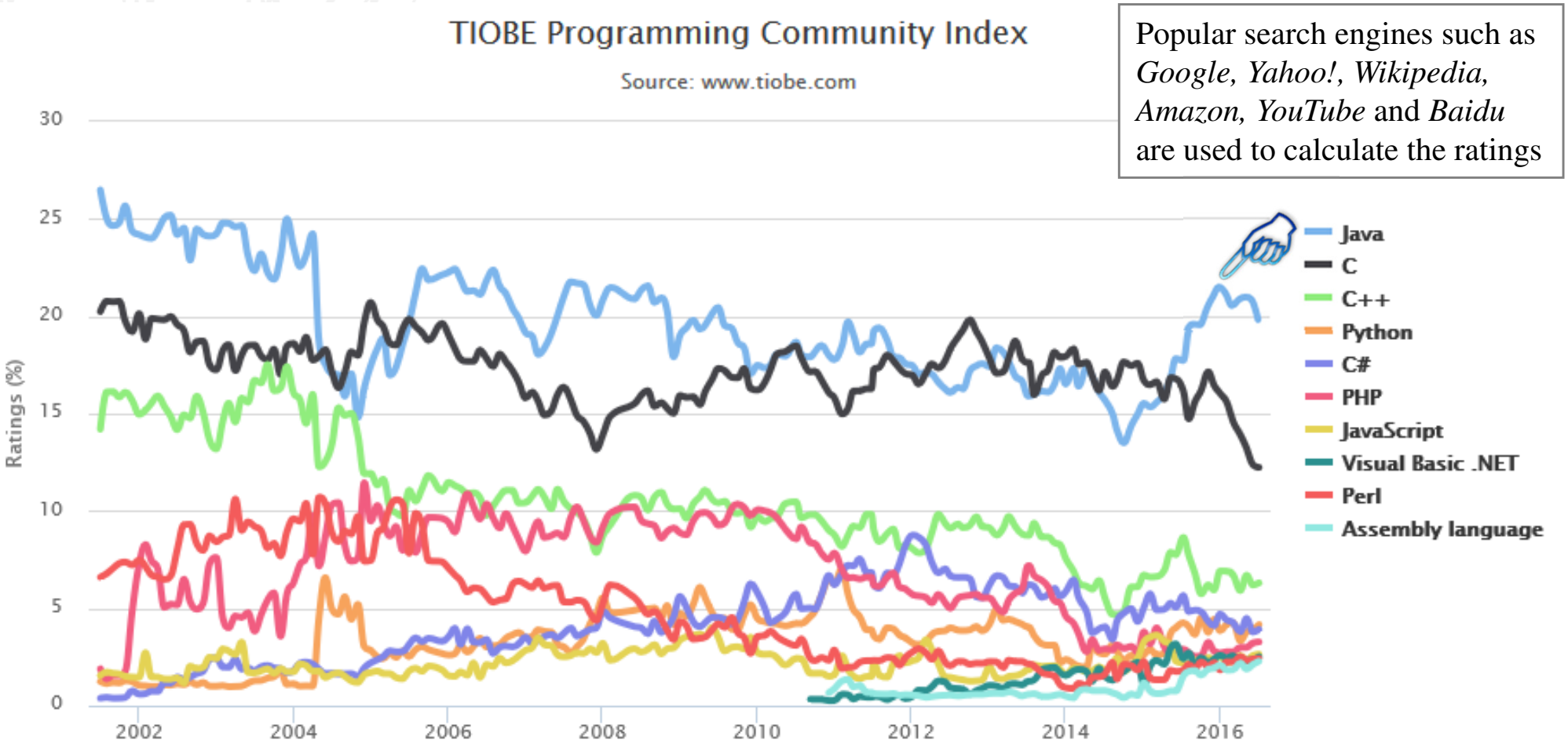
Why Java?

- General-purpose language
  - Suitable for a wide range of applications
- **Supports the most advanced software development concepts**
  - class-based, object-oriented, concurrent
- Architecture-neutral, portable
  - write once, run anywhere
- Most popular
  - Java is everywhere
    - 97% of enterprise desktops run Java
    - 3 billion mobile phones run Java
    - 100% of Blu-ray disc players ship with Java
    - 5 billion smart cards run Java Card applets
    - 125 million TVs run Java
    - Java powers set-top boxes, printers, Web cams, games, car navigation systems, lottery terminals, medical devices, parking payment stations, and more
  - 9 million programmers/developers
  - Books, tutorials, exercises, compilers, environments are just as numerous as stars

If one is a good master of one programming language, one would have few difficulties to pick up another quickly

TIOBE Programming Community Index

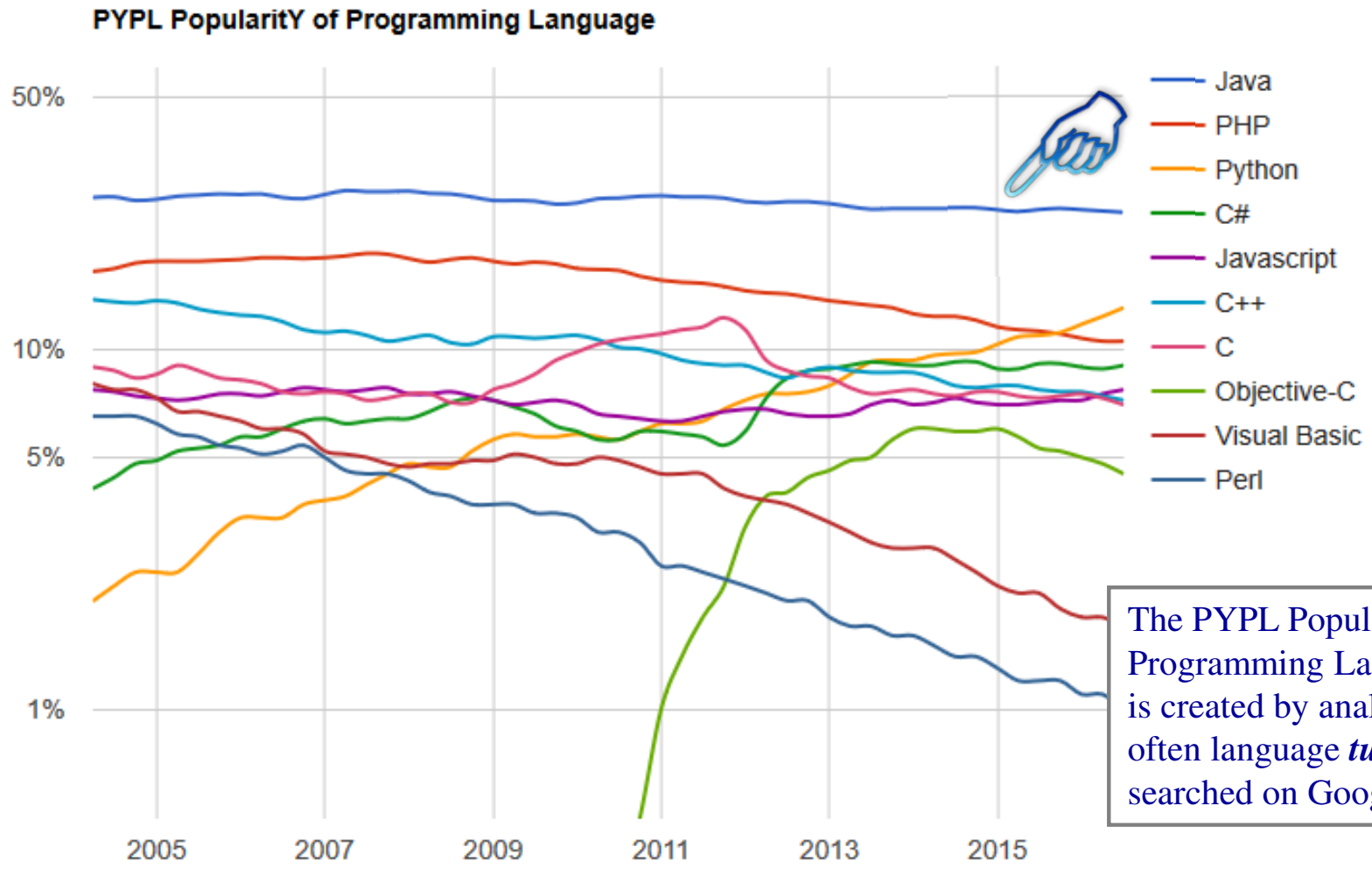
Long term trends for top 10 programming languages



Popular search engines such as *Google, Yahoo!, Wikipedia, Amazon, YouTube* and *Baidu* are used to calculate the ratings

Viewed in July 2016 at www.tiobe.com/tiobe_index

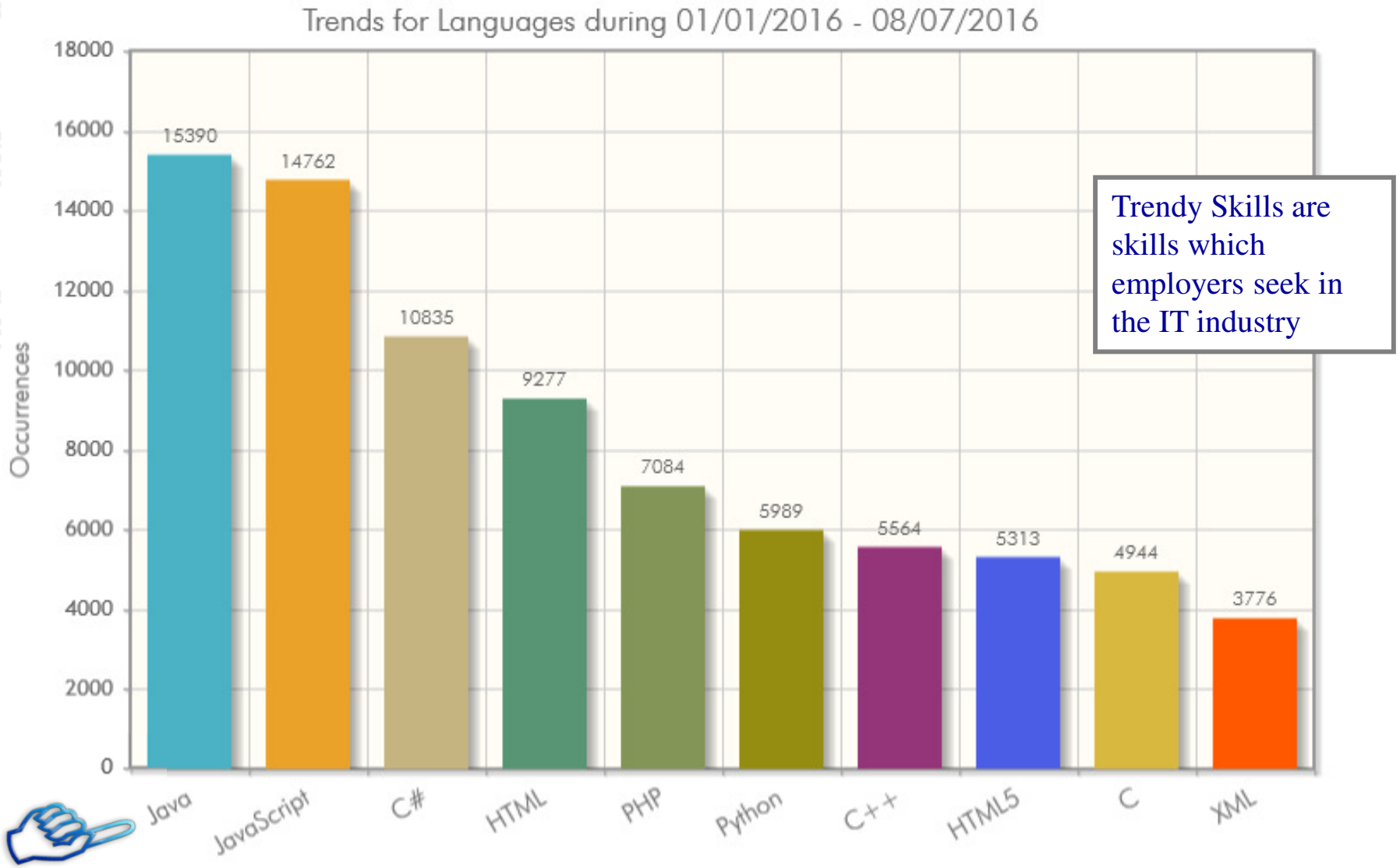
Programming Language Index



The PYPL PopularitY of Programming Language Index is created by analysing how often language ***tutorials*** are searched on Google

Viewed in July 2016 at http://pypl.github.io/PYPL.html

Trendy Skills

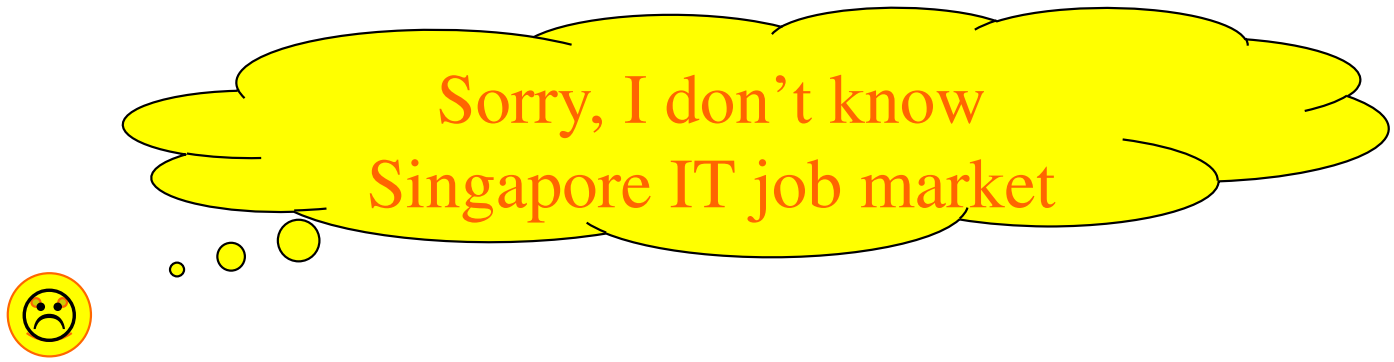


Trendy Skills are skills which employers seek in the IT industry

Viewed in July 2016 at trendyskills.com

Australian IT job market

Keywords	2013 Job Title: Developer		2014 Category: IT		2015 Category: IT		2016 Category: IT	
Java	80	47.6%	60	56.1%	355	66.6%	229	55.4%
C#	76	45.2%	34	31.8%	158	29.6%	173	41.8%
Objective C	12	7.1%	13	12.1%	20	3.8%	11	2.7%



Programming

- Programming is a problem-solving activity
  - A program tells the computer how to solve a specific problem
  - A problem can be broken down into a set of sub-problems
    - There are many ways how a problem can be subdivided into sub-problems
    - The subdivision affects the program implementation
- Programming is not very difficult, but time-consuming
  - The most challenging part about programming is to match subdivision of the problem with the program design methodology
  - Computers cannot guess what problem you are solving. They simply follow your program instructions even though they may be wrong
  - A lot of time will be spent to figure out why the computer does not do what you expect it to do - debugging

Remember? *Brilliant Humans beings are incredibly slow and inaccurate*

The subject

- Objective
  - Learn the fundamental principles of programming
    - *Object-oriented view* of *problem analysis and solving*
- Learning Outcomes
  1. Create and manipulate *data types* and *structures*.
  2. Design and implement solutions using *classes*; implement the *behaviour of objects* in a structured way.
  3. Understand and apply the *syntactic and semantic rules* of an object-oriented programming language.
  4. Illustrate an understanding of *tools and techniques* for *program testing*.
  5. Illustrate an understanding of the *concepts* involved in *compilation, linking and execution*

This subject is not just about Java, but you will learn programming in Java

Topics covered by CSIT111

1. Object-oriented design methodology
2. Java language basics
   - Data types and structures
   - Classes and objects
   - Object-oriented programming
     - object integration, Java libraries and class design
3. Java development environment
   - Concepts of compilation/linkage/execution
   - Program testing, maintaining and debugging

Topics not covered by CSIT111

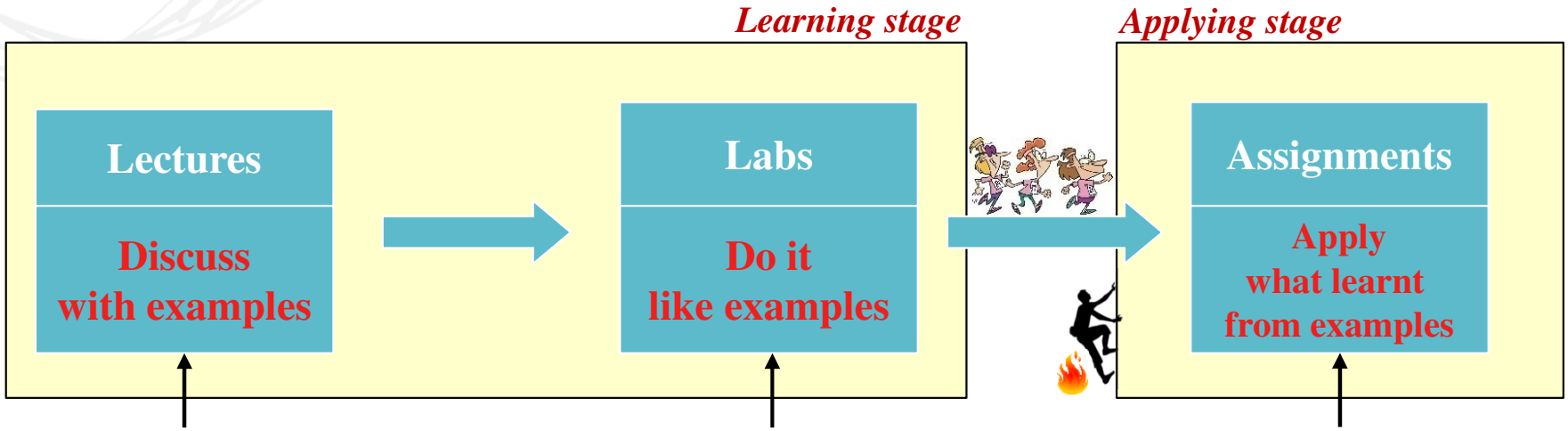
- Java Virtual Machine

  - Java is more than just a programming language

- Java platform and technology
  - Relationship between classes – inheritance and polymorphism
  - Concurrency (multi-threads)
  - GUI (Graphic User Interface) and graphics
  - Networking
  - Database connectivity
  - Enterprise, Micro (Card and TV) and Embedded Java

Learning programming

- Learn by ***doing***
  - ***If you want to learn programming, you must "do" programming***
  - ***Spend sufficient time doing it***

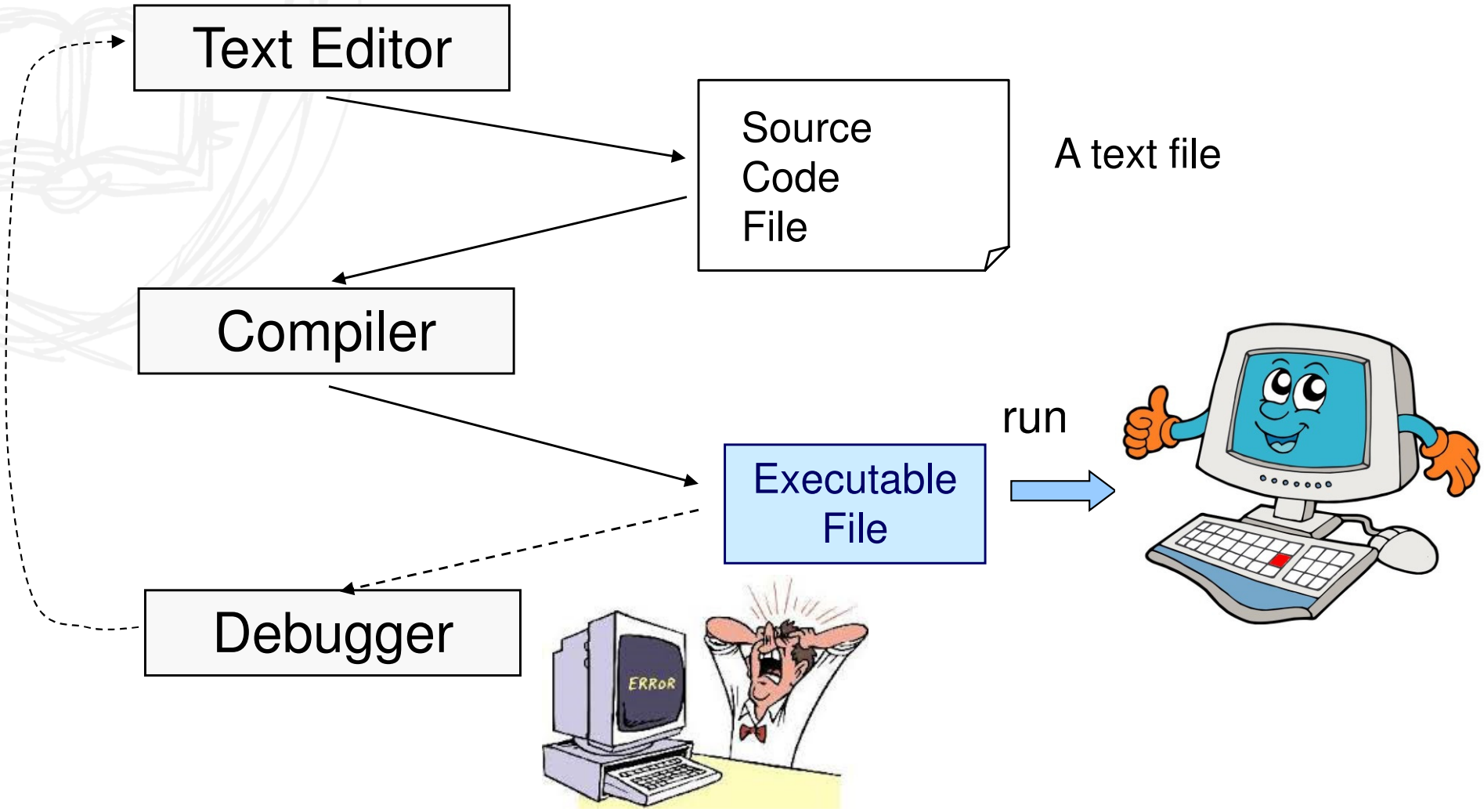


- You should listen to what the lecturer says about what on the presentation slides;
- Slides are not for you to read - books are

- You must do exercises if you want to learn programming.
- If you do not spend sufficient time to actually write actual code, you are not learning programming.

- When you are ready, you will find assignments not very difficult to complete

Programming Process



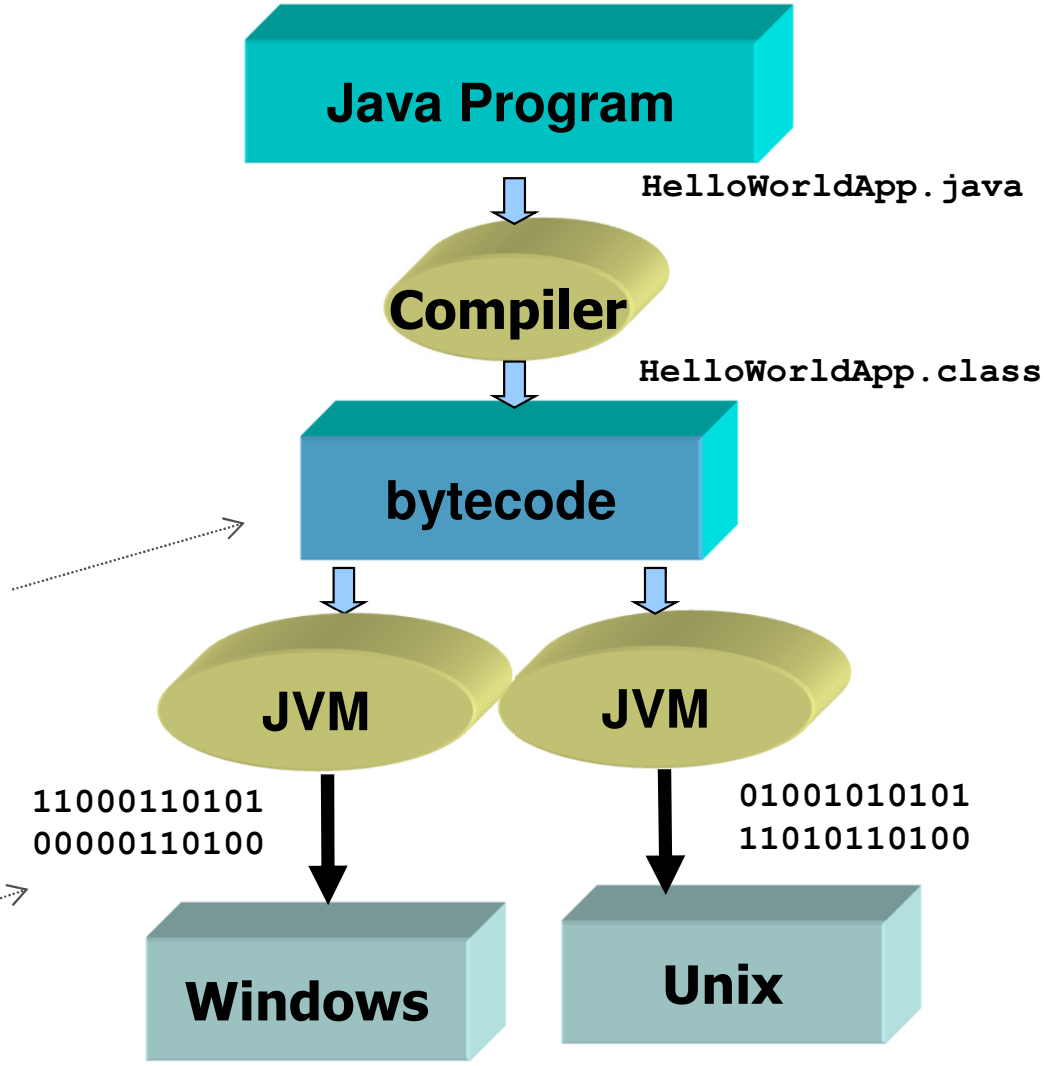
Java Programming Process

- Implement a program
  - Use any text editor to write it
  - Save text into a file
    `HelloWorldApp.java`
- Compile a program: compiler
  - **`javac HelloWorldApp.java`**

Java bytecode is hardware platform independent

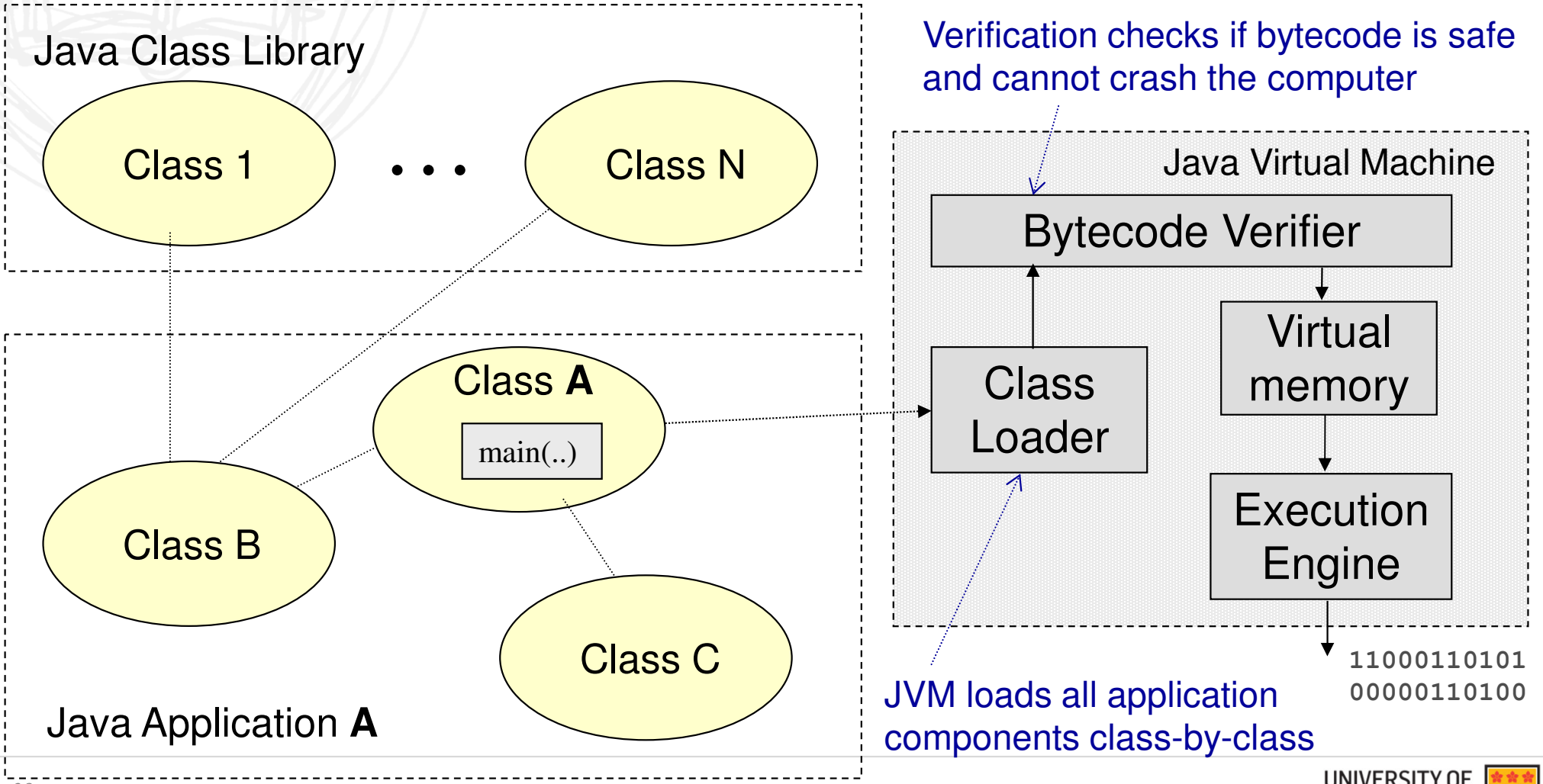
- Executing a program: interpreter
  - **`java HelloWorldApp`**

JVM translates bytecode into hardware platform dependent microprocessor instructions



Java Program Execution

- The basic structural unit of a Java program is a class
- Instances of classes placed in memory become objects



1. Write your program

```
/**
 * The HelloWorldApp class implements an application
 *  that displays "Hello world!" to the standard
 *  output.
*/
class HelloWorldApp{
   public static void main(String[] args){
        // Display "Hello world!"
        System.out.println("Hello world!");
   }
}
```

A predefined object

Text editors for programmers:

- Pure text based (no graphics, etc)
- Syntax highlighting
- Note: word processors are not suitable source code editors
  - ~~MS Word~~
  - ~~FrameMaker~~

OS	Editors
Windows	Notepad++
Linux (desktop)	gEdit, nEdit
Linux/Unix/Mac (CLI)	pico/nano (no syntax highlighting) vim/emacs (not for beginners)
Mac	TextMate

1. Write your program

- Source code must be properly formatted using whitespace
- Chaotic indentation makes a program code messy

```
class HelloWorldApp {public static void main(String[]
args){System.out.println("Hello world!");}}
```

Bad example!

```
class HelloWorldApp
{public static void main(String[]
args){System.out.println("Hello world!");}}
```

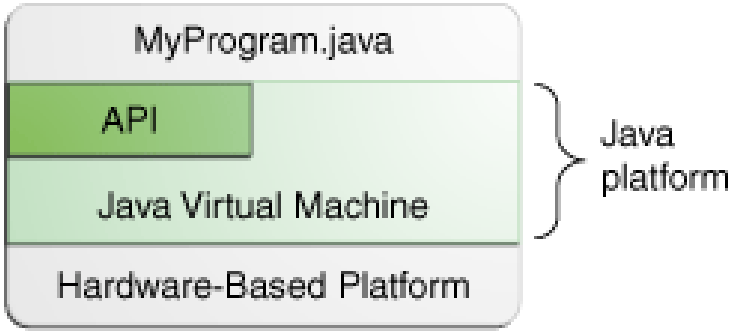
Such code can be compiled, but you will lose marks for submitting such code of bad style resulting in poor readability

2. Compile and run your program

- To compile your program you need

  Java Development Kit (JDK) that includes

  - `javac` Java compiler
  - Java Virtual Machine
  - Java APIs
  - other components and tools



- To run your program you need

  Java Runtime Environment (JRE) that is a subset of JDK

  - `java` Java application loader
  - Java Virtual Machine
  - Java Class Library
  - other components

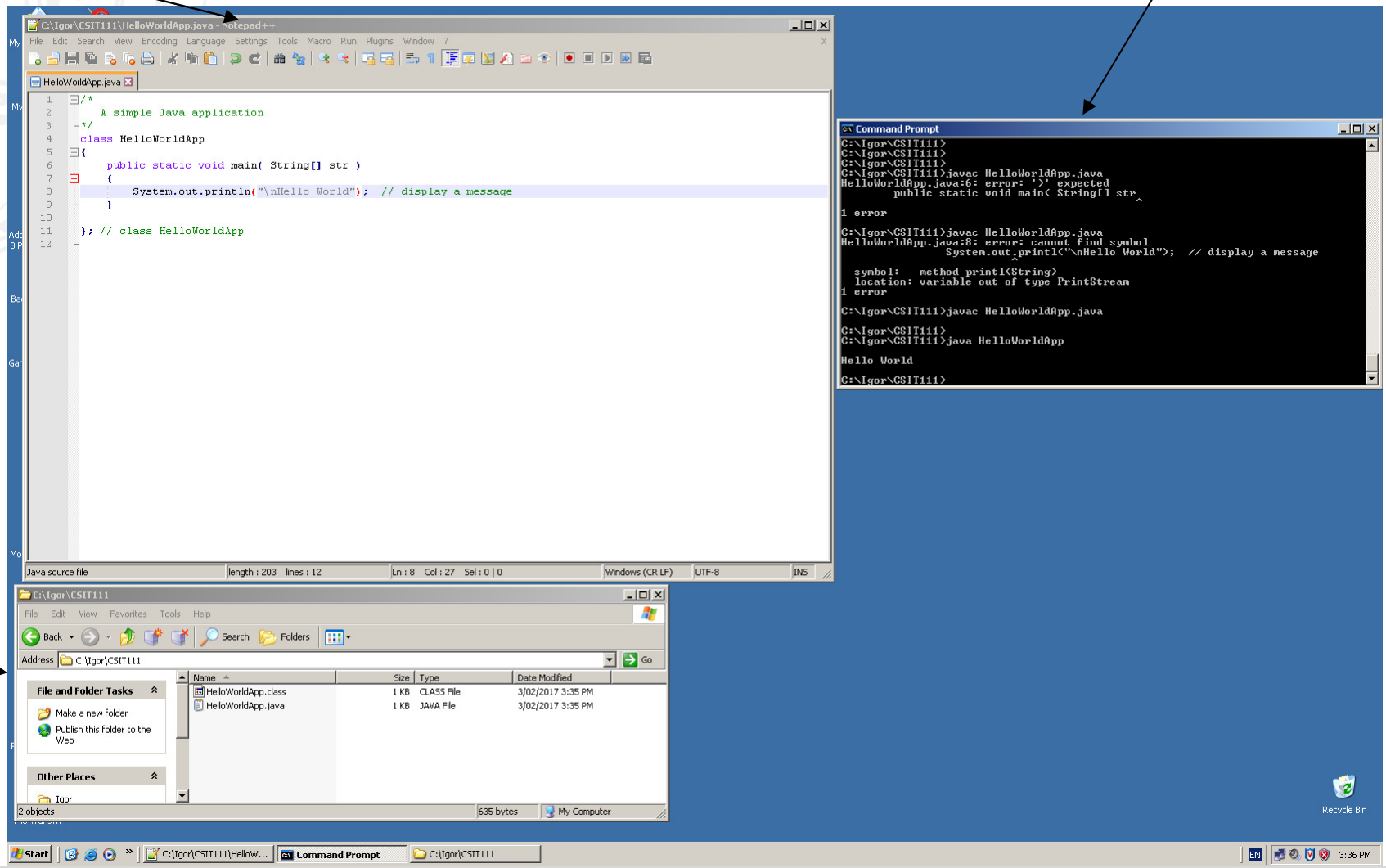
There is no need to install JRE
if you have installed JDK

Basic development environment

Notepad++

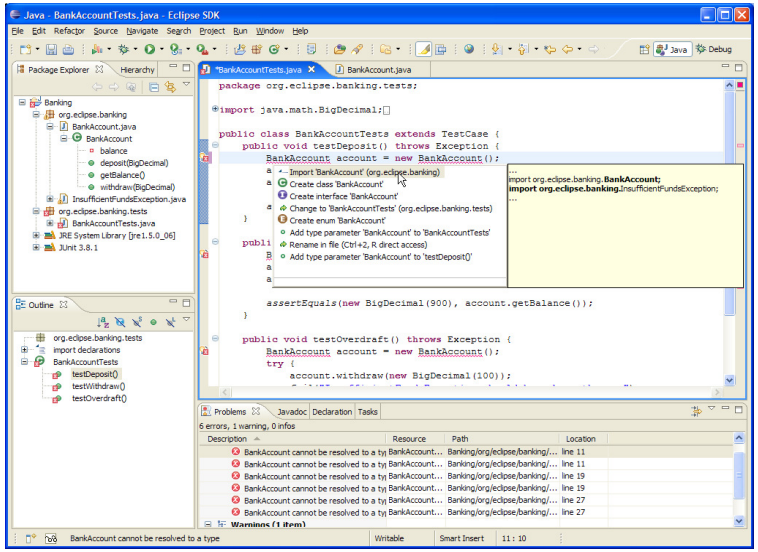
Command Prompt

Windows Explorer



Integrated Development Environment

- Although JDK provides all tools needed for compiling, debugging, running and documenting your program, you may find it more convenient to use an application that integrates all tools and provides a convenient graphical user interface



- Integrated Development Environments (IDE)
  - Text editor + Compiler + debugger + code manager
  - Graphic user interface (window-based)
  - All platforms (Windows/Linux/Mac)

IDE	Remarks
BlueJ	Beginners
NetBeans	Full featured
Eclipse	Industrial strength
IntelliJ IDEA	Android based on

A complex IDE does not help beginners to learn programming as it increases productivity only when you know what to write.
You should first focus on what to write

Java versions

- Java has undergone substantial changes since it was introduced in 1995
  - syntax
  - class library
  - performance

1995	1997	1998	2000	2002	2004	2006	2011	2014
1.0	1.1	1.2	1.3	1.4	5	6	7	8

The latest release is Version 8 Update 171

- `java -version` command checks what version is installed on your computer

Java version used for CSIT111

- CSIT111 is based on Java Standard Edition 8 (Java SE8)
  - This version supports all features needed to develop desktop and server applications.
- Java SE8 adds support of a new programming concept - functional programming (not covered by CSIT111)
- The Java Enterprise Edition (Java EE) is focused mostly on developing large-scale, distributed networking applications (not covered by CSIT111)

1. Download JDK SE from www.oracle.com and install on your computer
2. Download and install Notepad++ source code editor

First Program

```
/**
 * The HelloWorldApp class implements an application
 *  that displays "Hello world!" to the standard
 *  output.
*/
class HelloWorldApp{
  public static void main( String[] args   {
        // Display "Hello world!"
        System.out.printl ("Hello world!");
    }
}
```

a bug
) is missing
a bug
n *is missing*

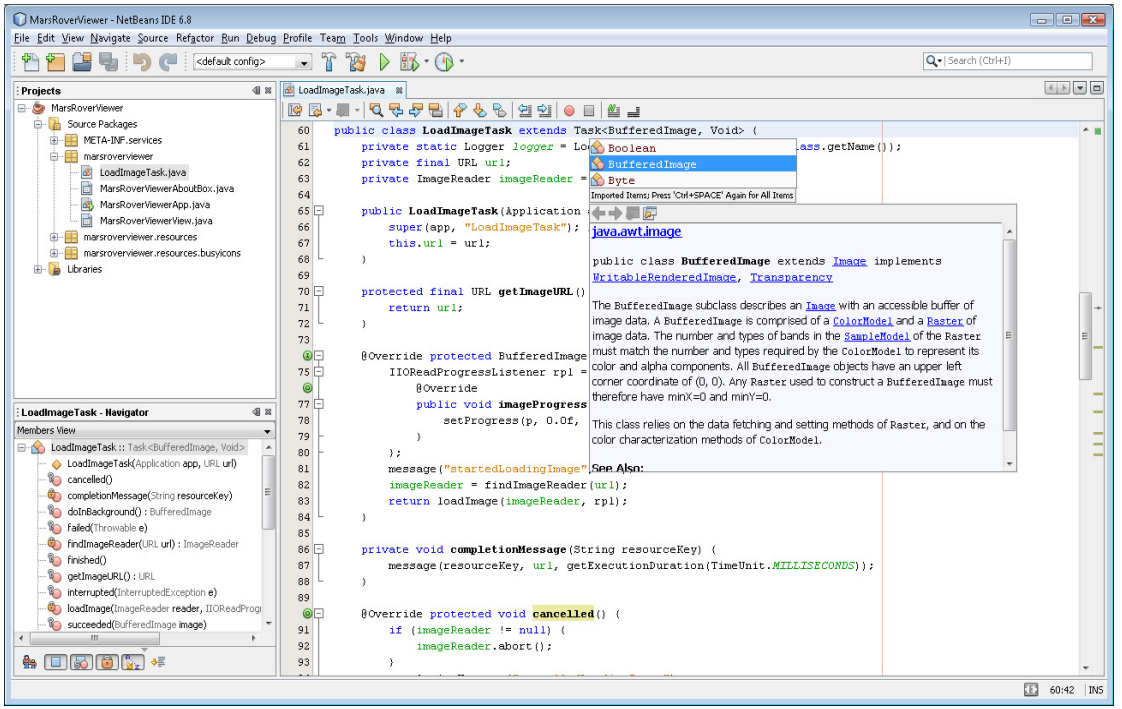
Compilation errors

```
C:\Igor\CSIT111>javac HelloWorldApp.java
HelloWorldApp.java:6: error: ')' expected
        public static void main( String[] str
                                              ^
1 error

C:\Igor\CSIT111>javac HelloWorldApp.java
HelloWorldApp.java:8: error: cannot find symbol
                System.out.printl("\nHello World");
                           ^
  symbol:   method printl(String)
  location: variable out of type PrintStream
1 error

C:\Igor\CSIT111>javac HelloWorldApp.java

C:\Igor\CSIT111>
C:\Igor\CSIT111>java HelloWorldApp

Hello World

C:\Igor\CSIT111>
```

- Everything in a Java program must be inside a class
- Class – a container for the program logic that defines the behaviour of which all Java applications are built

NetBeans IDE

- Once you get familiar with the basics of Java programming, you can try using NetBeans IDE
- Although it is an advanced development tool, it is not hard to learn how to use its basic features
  - create a Java project
  - edit source code files
  - compile
  - debug
  - test run
- It supports many other things which you don't need at this stage

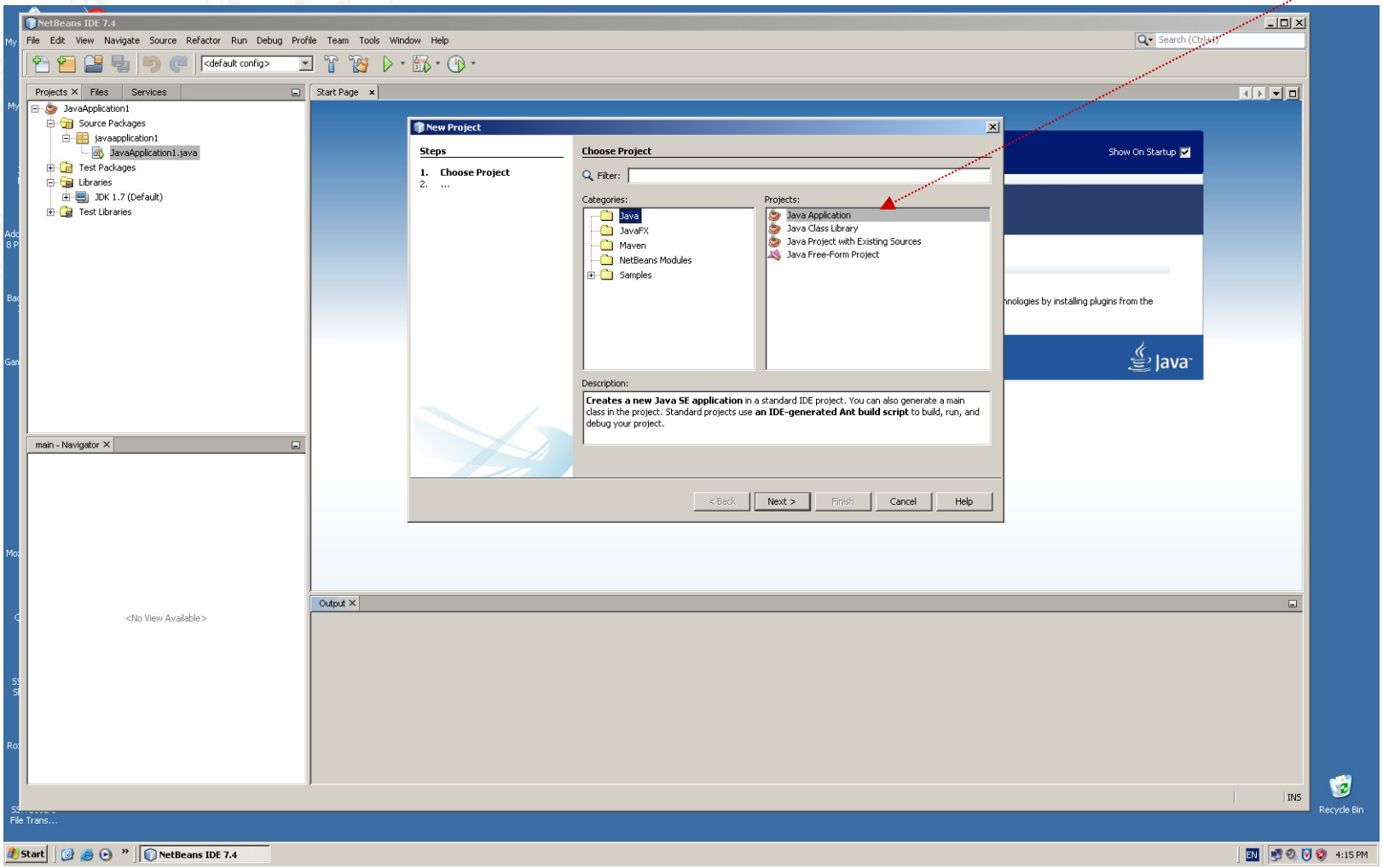


- NetBeans IDE does not have its own Java compiler. It uses JDK that should be installed before you install NetBeans

NetBeans IDE

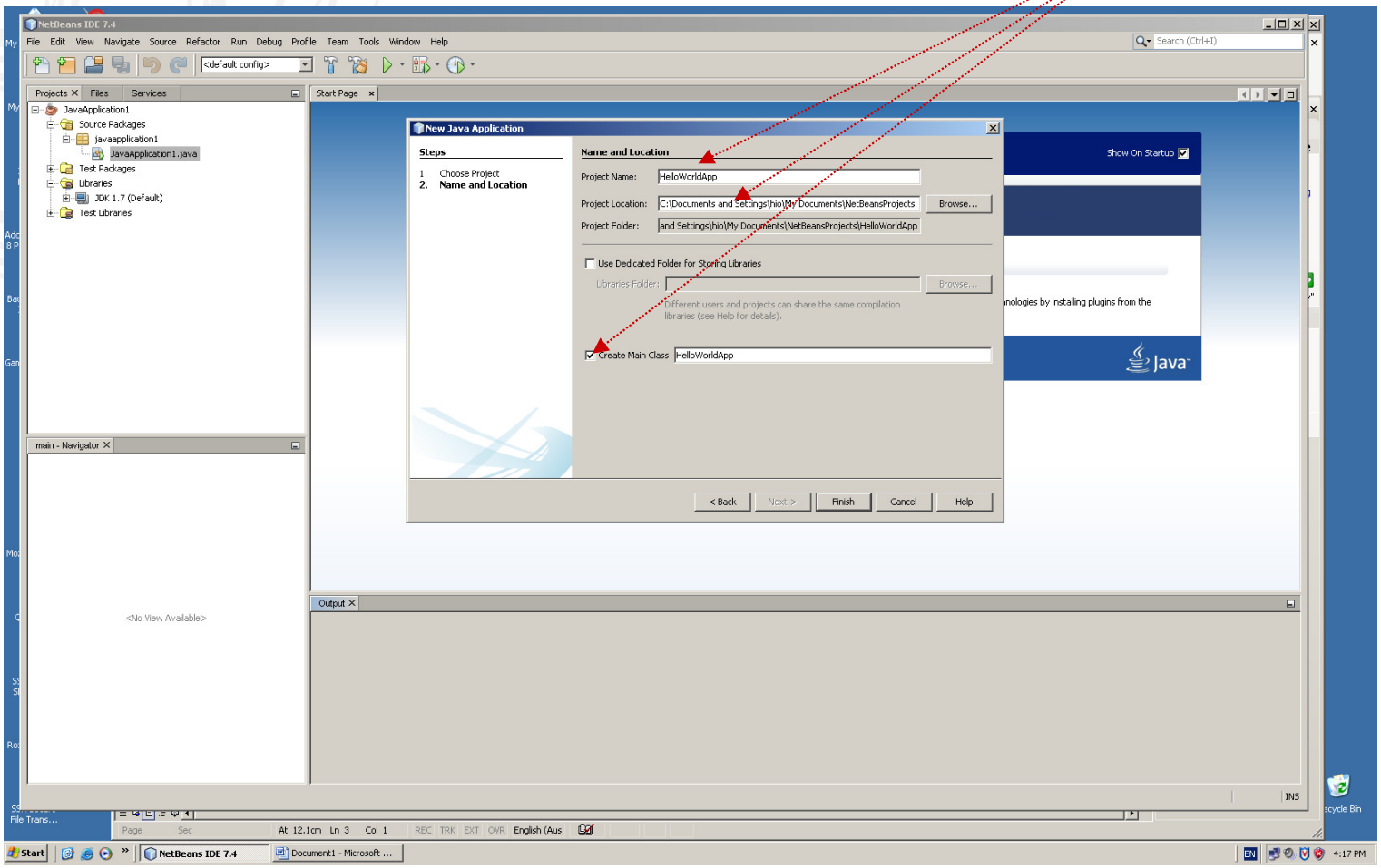
1. Create a new project or open an existing project

Select **Java Application**



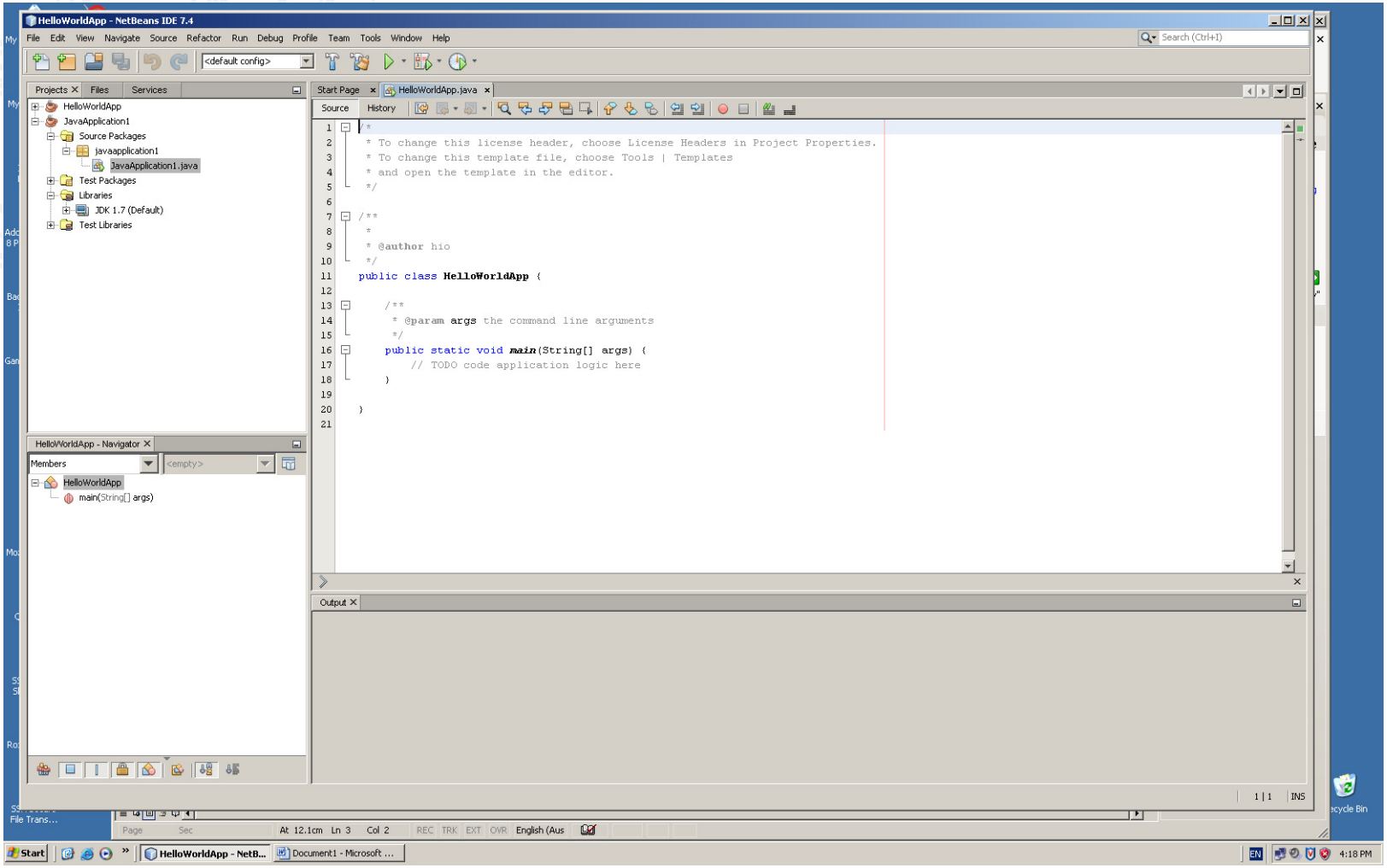
NetBeans IDE

2. Specify the project name, location of its folder and tick Create Main Class. Then press Finish.



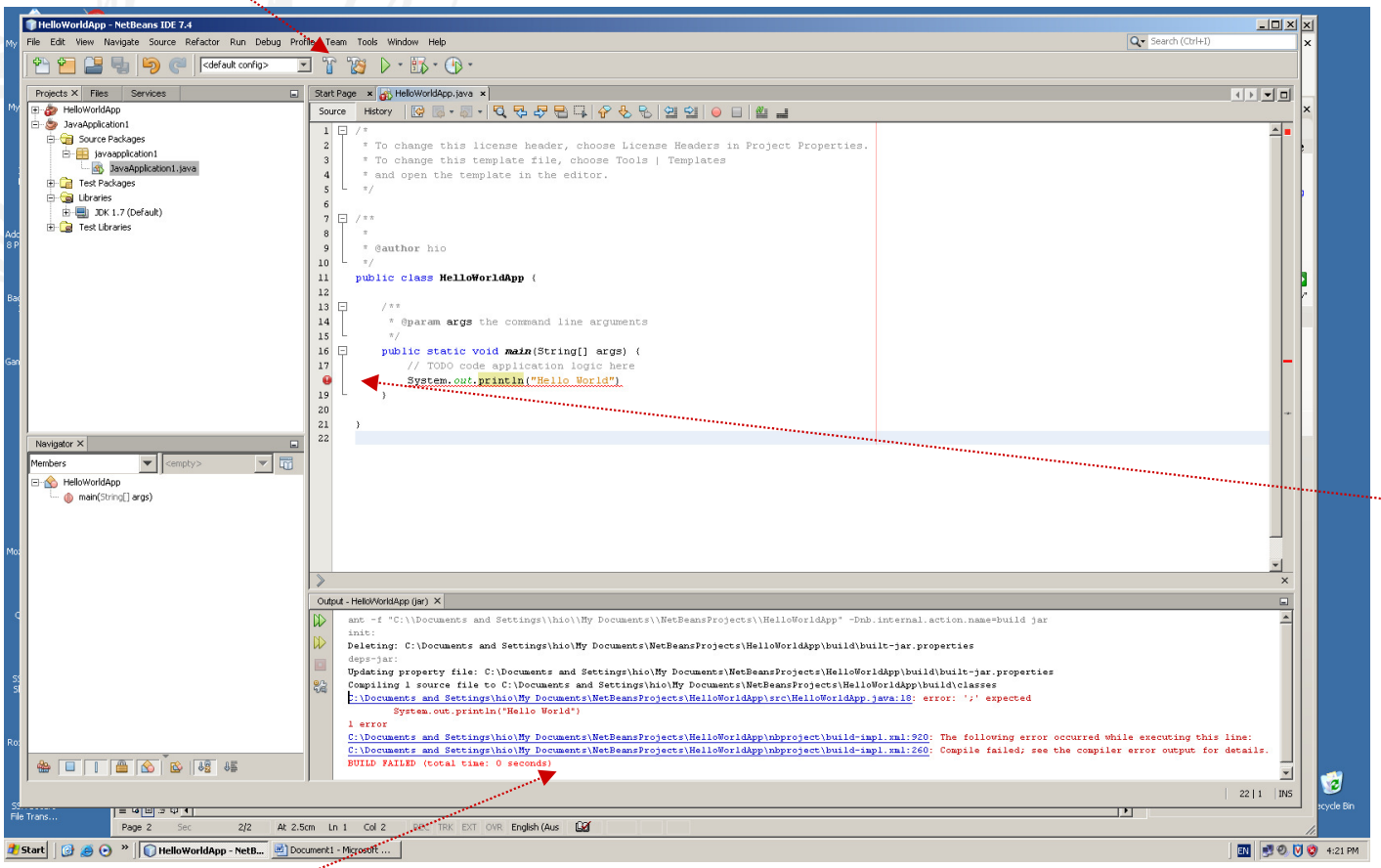
NetBeans IDE

3. A template source code will be generated that you can edit



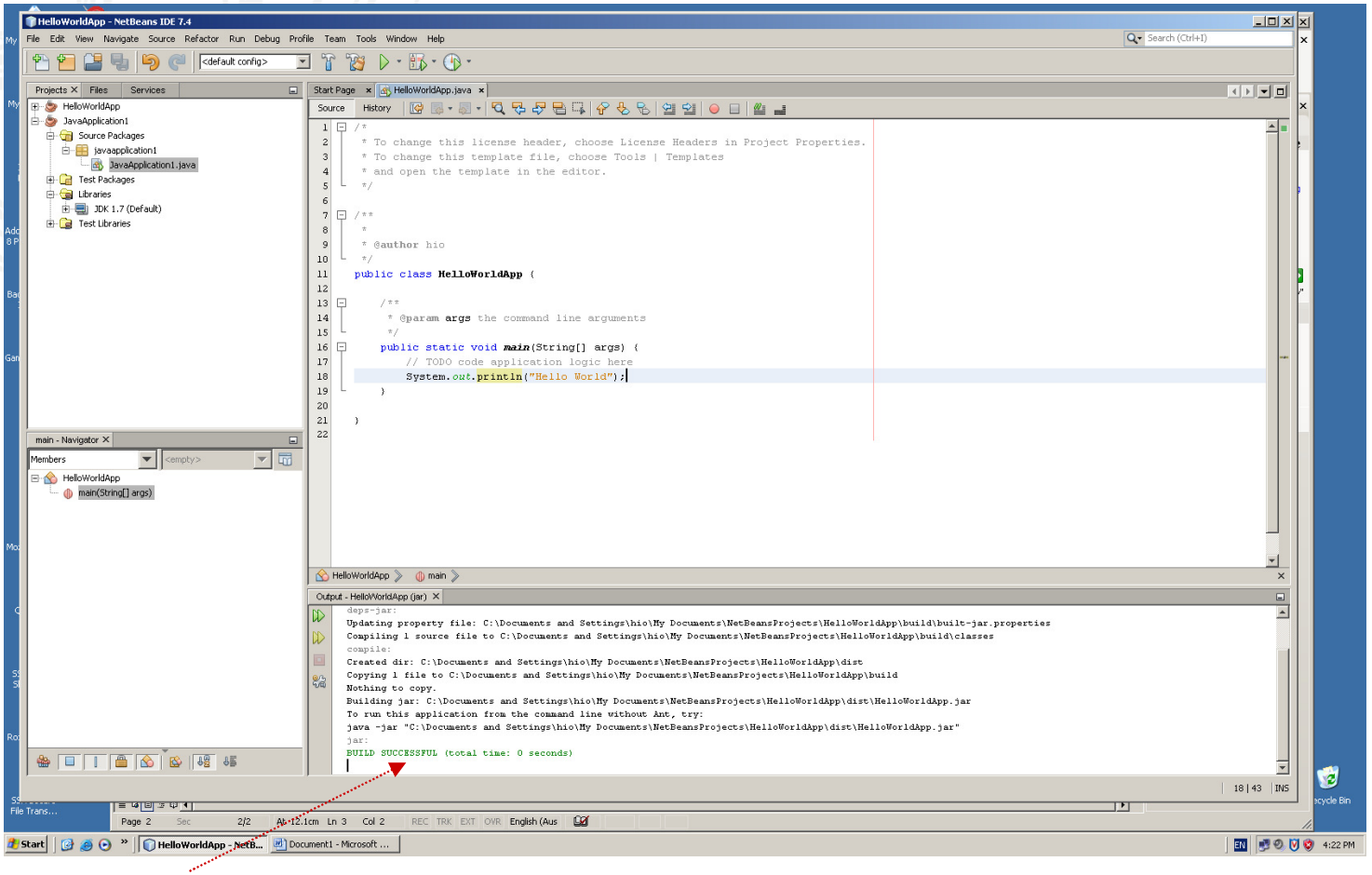
NetBeans IDE

4. When you finish editing, you can compile you program by pressing **Build Project** button



NetBeans IDE

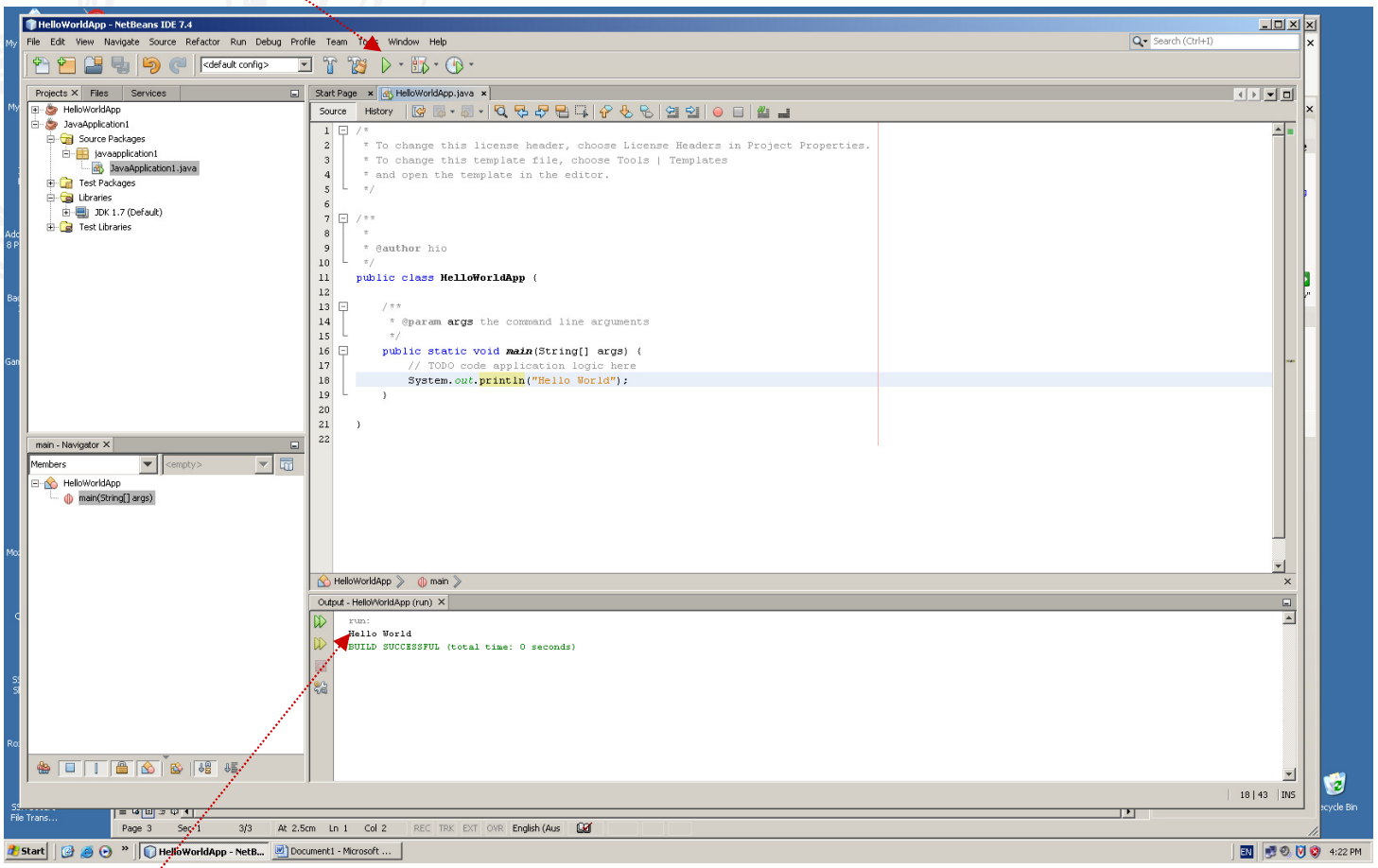
5. When you fix reported bugs, you can re-compile you program by pressing **Build Project** button again



Build Successful message

NetBeans IDE

5. After successful compilation you can run your application by pressing **Run Project** button



`Hello World` message printed by the running application

Suggested reading

Java: How to Program (Early Objects), 11th Edition

- Preface
- Before you begin
- Chapter 1
  - 1.2 Hardware and software
  - 1.4 Machine languages
  - 1.6 Operating systems
  - 1.9 Java development environment