Lecture#1
Object Oriented Programming (JAVA)

Dr. Abu Nowshed Chy

Department of Computer Science and Engineering

University of Chittagong

December 11, 2023

Faculty Profile

Course Information

Course Materials:

1. Kay S. Horstmann: "*Core Java: Volume I – Fundamentals*", 11th Edition

2. Paul Deitel and Harvey Deitel: "*Java: How to Program*", 10th Edition, Pearson Education Asia.

3. E. Balagurusamy : "*Programming with Java-A Primer*", 3rd Edition, Tata McGraw-Hill Publishing Ltd.

4. Other specific materials and lecture slides

Why Programming!!!!

To Solve Real-World Problem …

Problem Solving

Golden Rules of Programming →

- Think before you code
- Always choose the simplest solution that is fast
- Code carefully rather than fast
- Dedication

Problem Solving

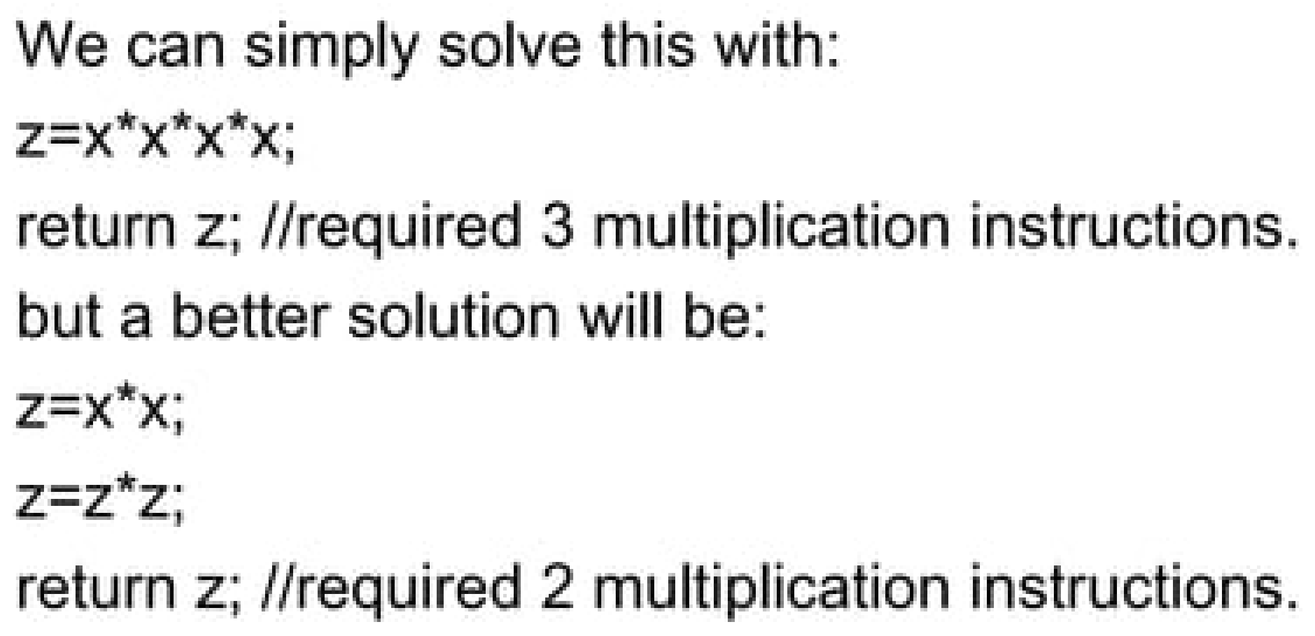
Think before you code →

- The purpose of writing a program is to solve a problem
- Solving a problem consists of multiple activities:
  - Understand the problem
  - Design a solution
  - Consider alternatives and refine the solution
  - Implement the solution
  - Test the solution
- These activities are not purely linear

Problem Solving

Choose the simplest but fast solution →

- Say you want to write a function to calculate x^4.



We can simply solve this with:

```
z=x*x*x*x;
return z; //required 3 multiplication instructions.
```

but a better solution will be:

```
z=x*x;
z=z*z;
return z; //required 2 multiplication instructions.
```

Problem Solving

Choose the simplest but fast solution →

```
for(i=0;i<100;i++)
    if(i<50)
        a[i]=…
    else
        b[i]=…
```

Problem Solving

Choose the simplest but fast solution →

```
for(i=0;i<100;i++)
      sum=sum+i;
```

College Level Math Knowledge

- Set
- Real Numbers
- Polynomial and Polynomial Equations
- Matrix and Determinant
- Permutations and Combinations
- Binomial Theorem
- Summations and Series (Induction)
- Coordinates
- Straight Lines

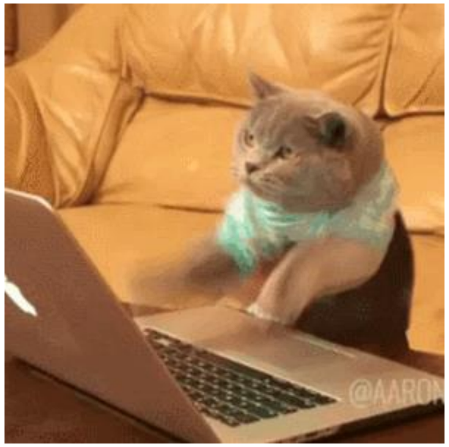
College Level Math Knowledge

- Circle
- Conics
- Vector
- Basic Trigonometric Formula
- Integration and Differentiations rules
- Probability
- Bayes Theorem

Problem Solving

Code carefully rather than fast →

- ❖ Easy to make mistakes when coding fast
- ❖ Mistakes are hard to find and take long to fix
- ❖ Really try to avoid making mistakes
- ❖ Coding carefully is actually faster in the end!



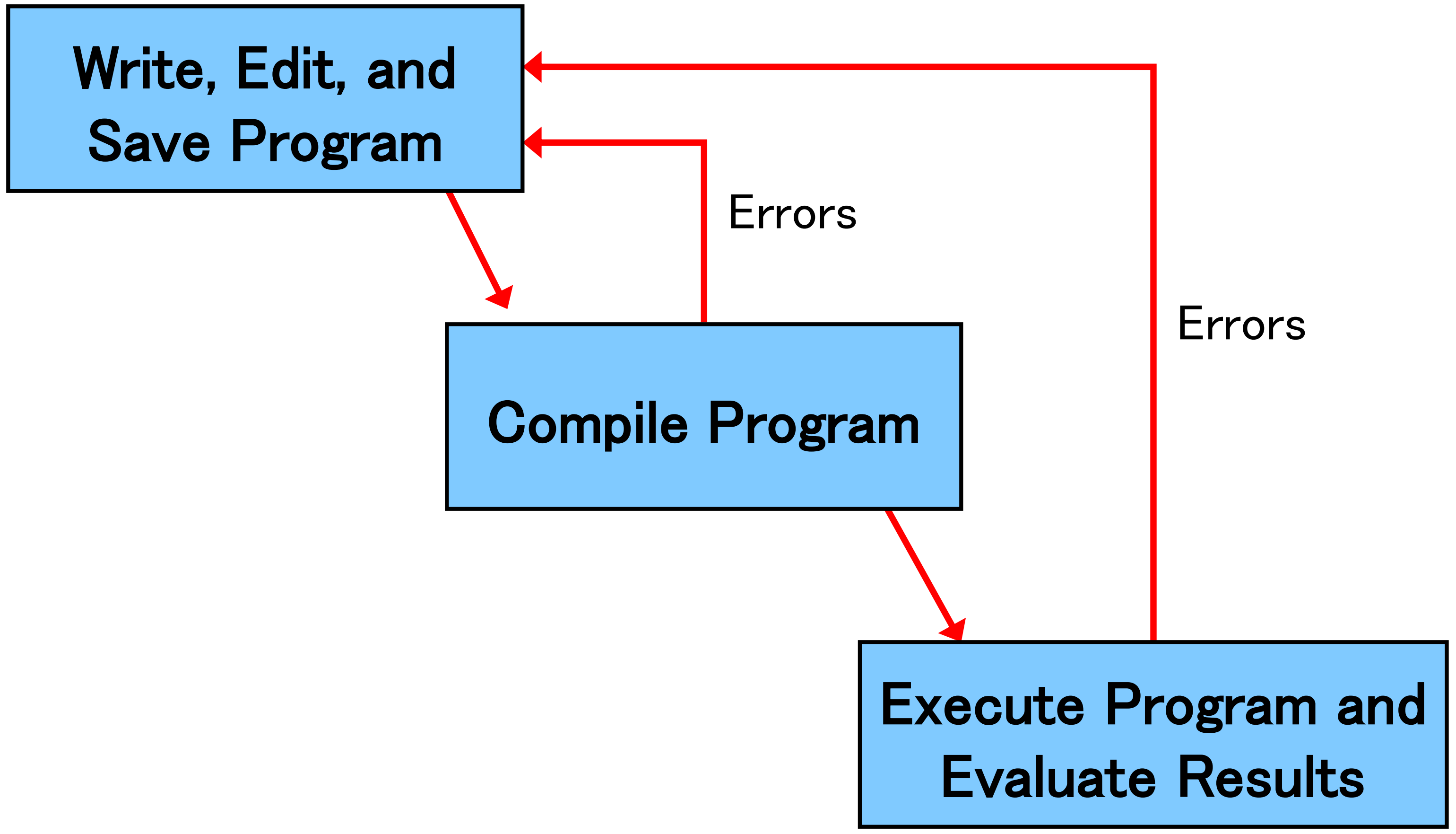
Problem Solving

Code carefully rather than fast →

How to avoid mistakes?

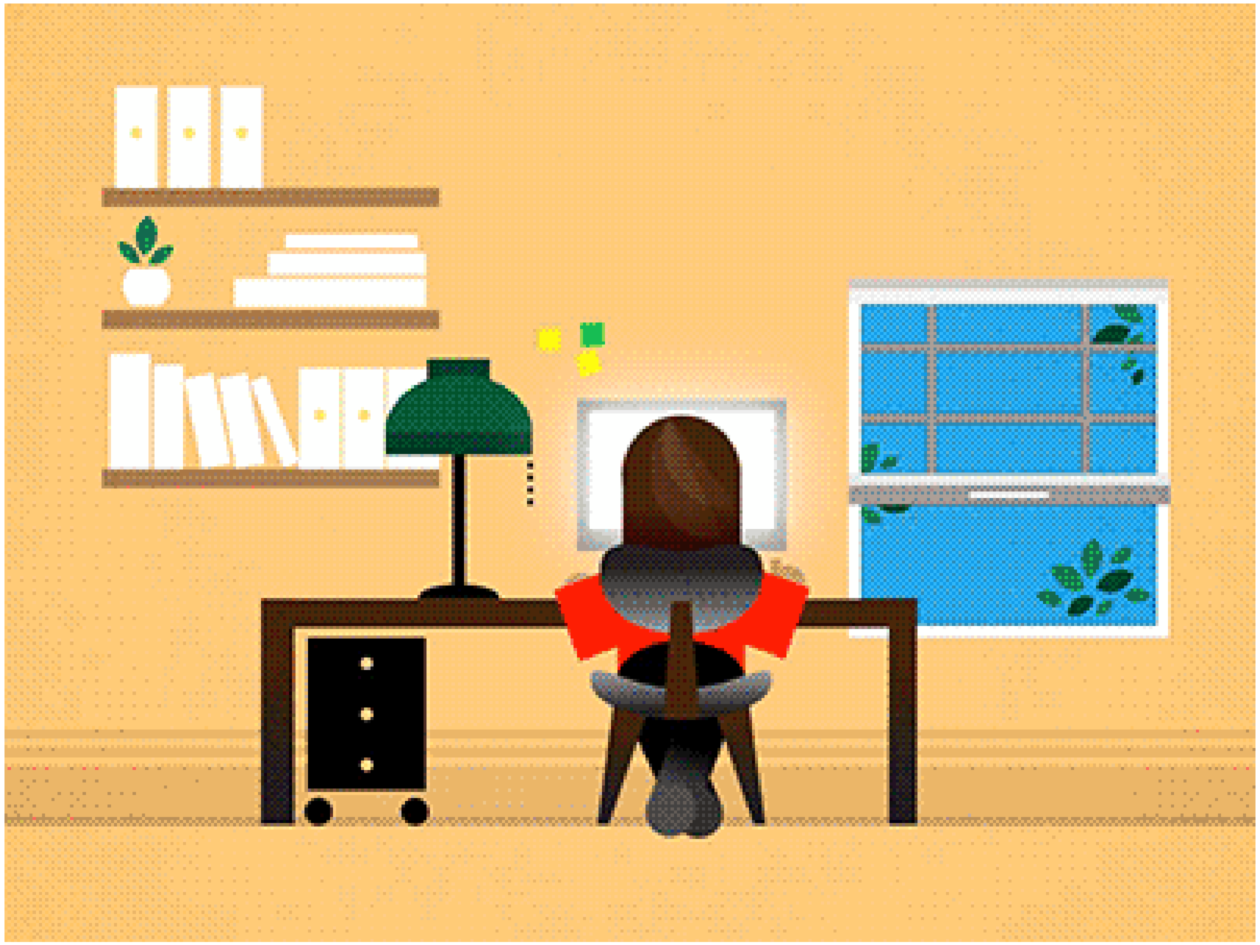
- ❖ No matter how careful, you will make mistakes
- ❖ It helps to know common mistakes
- ❖ Helps finding, correcting, and avoiding mistakes

Basic Program Development


Write, Edit, and Save Program
Errors
Errors
Compile Program
Execute Program and Evaluate Results

Don’t Do That !!!!

Be Passionate About Programming!



But Not this Type!!!

Problem Solving

Modular programming→

- The key to designing a solution is breaking it down into manageable pieces
- When writing software, we design separate pieces that are responsible for certain parts of the solution
- An $\boxed{\textit{object-oriented approach}}$ lends itself to this kind of solution decomposition
- We will dissect our solutions into pieces called objects and classes

Course Objective

- To provide knowledge of fundamental concepts in OOP
- Develop an understanding of OOP design artifacts
- Familiarize with the writing of computer programs to solve real-world problems using Java
- Design and implement object-oriented solutions

Java

It is a general purpose concurrent object oriented language, with a syntax similar to C and C++, but omitting features that are complex and unsafe.

Java

```
/**
* Hello World Application
* Our first example
*/
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World!"); // display output
  }
}
```

Java: Platform Independent

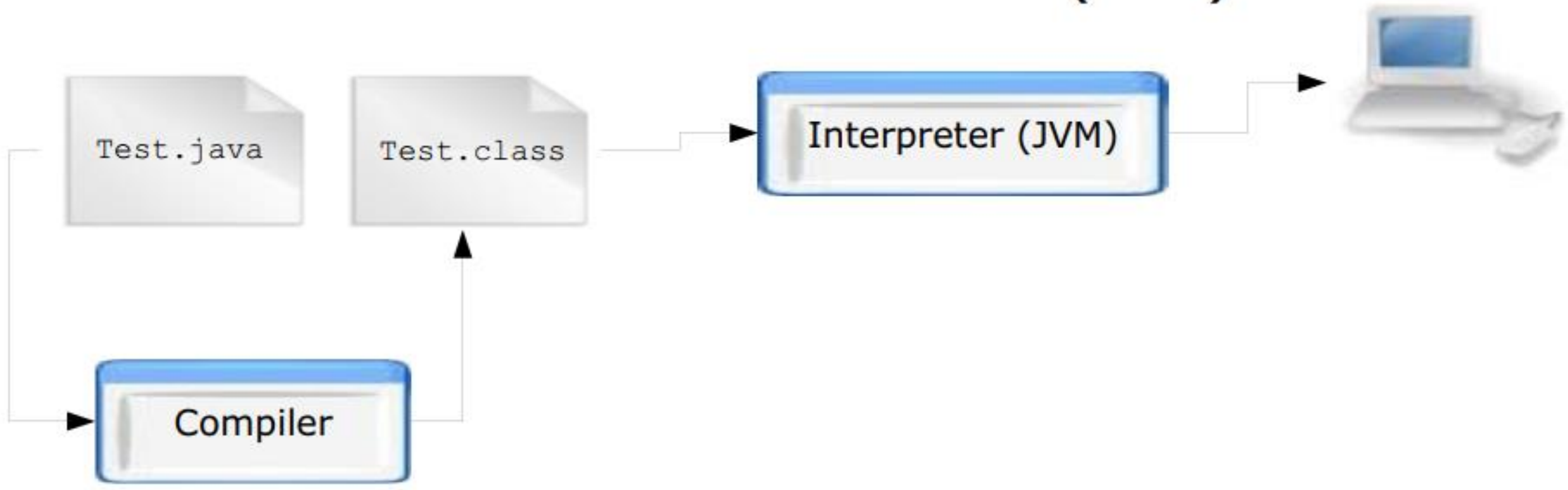
- Each type of CPU executes only a particular *machine language*
- A program must be translated into machine language before it can be executed
- A *compiler* is a software tool which translates *source code* into a specific target language
- Often, that target language is the machine language for a particular CPU type
- The Java approach is somewhat different

Java: Platform Independent

- The Java compiler translates Java source code into a special representation called *bytecode*
- Java bytecode is not the machine language for any traditional CPU
- Another software tool, called an *interpreter*, translates bytecode into machine language and executes it
- Therefore the Java compiler is not tied to any particular machine
- Java is considered to be *architecture-neutral*

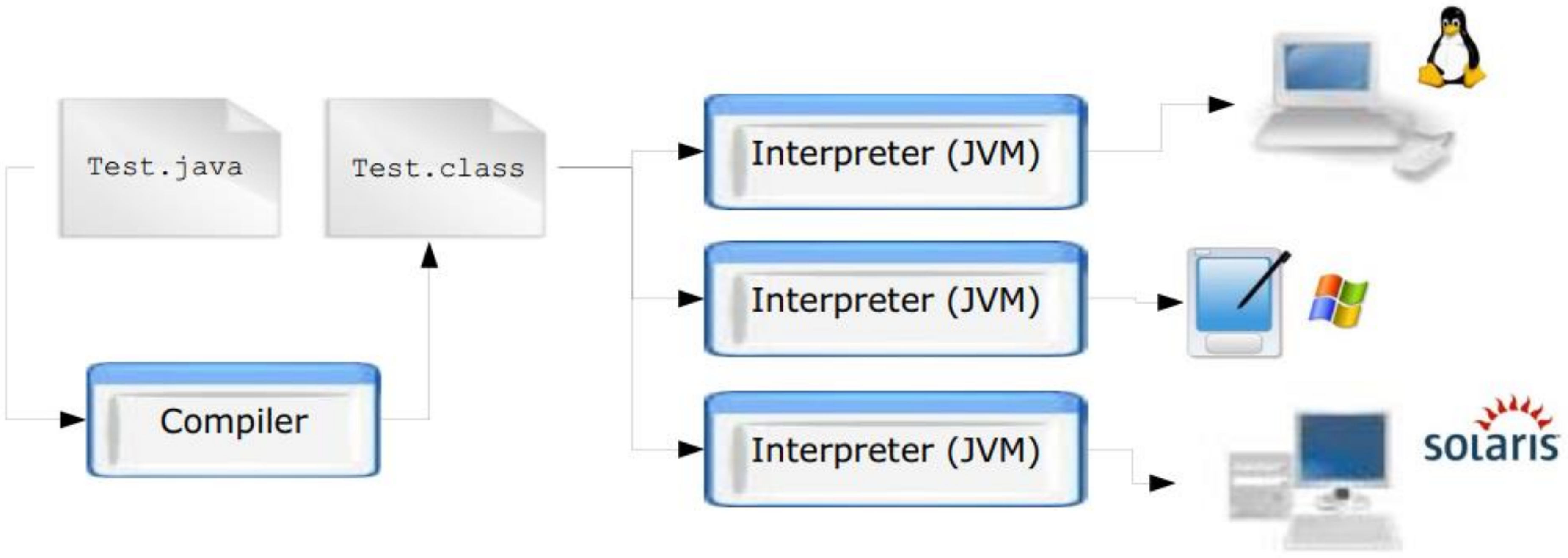
Java: Platform Independent

- Java programs are compiled to Java byte-codes, a kind of machine independent representation. The program is then executed by an interpreter called the Java Virtual Machine (JVM).



Java: Platform Independent

- The compiled code is independent of the architecture of the computer.
- The price to pay is a slower execution.



Thank You!