1. Overview of the Java Language

- **What Is the Java™ Technology?**

Java technology is:

• A programming language
• A development environment
• An application environment
• A deployment environment
• It is similar in syntax to C++.
• It is used for developing both applets and applications.

Java is an object-oriented programming language developed by Sun Microsystems. Originally the developers of Java intended to use C++ for their software development. But they needed a language that could execute on different sets of computer chips to accommodate the ever-changing consumer electronics market. So they decided to design their own language which would be independent of the underlying hardware.

It allows a user to receive software from a remote system and execute it on a local system, regardless of the underlying hardware or operating system. An interpreter and runtime are called the **Java Virtual Machine** which insulates the software from the underlying hardware.

Unlike more traditional languages, Java source code does not get translated into the machine instructions for a particular computer platform. Instead, Java source code (.java) is compiled into an intermediate form called bytecodes which are stored in a .class file. These bytecodes can be executed on any computer system that implements a Java Virtual Machine (JVM). This portability is perhaps one of the most compelling features of the Java language, from a commercial perspective. In the current era of cross-platform application development, any tool that allows programmers to write code once and execute it on many platforms is going to get attention.

2. Java Language Characteristics

- The portable, interpreted nature of Java impacts its performance. While the performance of interpreted Java code is better than scripting languages and fast enough for interactive applications, it is slower than traditional languages whose source code is compiled directly into the machine code for a particular machine. To improve performance, Just-In-Time compilers (JITs) have been developed. A JIT compiler runs concurrently with the Java Virtual Machine and determines what pieces of Java code are called most often.
- The bytecode portability is what enables Java to be transported across a network and executed on any target computer system. Java applets are small Java programs designed to be included in an HTML (HyperText Markup Language) Web document. HTML tags specify the name of the Java applet +and its Uniform Resource Locator (URL). The URL is the location on the

Internet at which the applet bytecodes reside. When a Java-enabled Web browser displays an HTML document containing an applet tag, the Java bytecodes are downloaded from the specified location and the Java Virtual Machine interprets or executes the bytecodes. Java applets are what enable Web pages to contain animated graphics and interactive content.

- Because Java applets can be downloaded from any system, security mechanisms exist within the Java Virtual Machine to protect against malicious or errant applets.

- Java is an object-oriented programming language, borrowing heavily from Smalltalk, Objective C, and C++. It is characterized by many as a better, safer C++. Java uses C++ syntax and is readily accessible to the large existing C++ development community.

- Java, however, does not drag along the legacy of C. It does not allow global variables, functions, or procedures. With the exception of a few primitive data types like integers or floating-point numbers, everything **in Java is an object**.

- **Object references are not pointers, and pointer manipulation is not allowed**. This contributes to the general robustness of Java programs since pointer operations tend to be particularly nasty and bug-prone. Java also manages memory itself, thereby avoiding problems with allocation and deallocation of objects. It does not allow multiple inheritance like C++ does, but supports another type of reuse through the use of formal interface definitions.

- Java is similar enough to C and C++ that it already feels familiar to most of the existing programming community. But it is different enough in important ways (memory management and cross-platform portability) that it is worth it for programmers to switch to a new language.

3. The Java Programming Language Basics

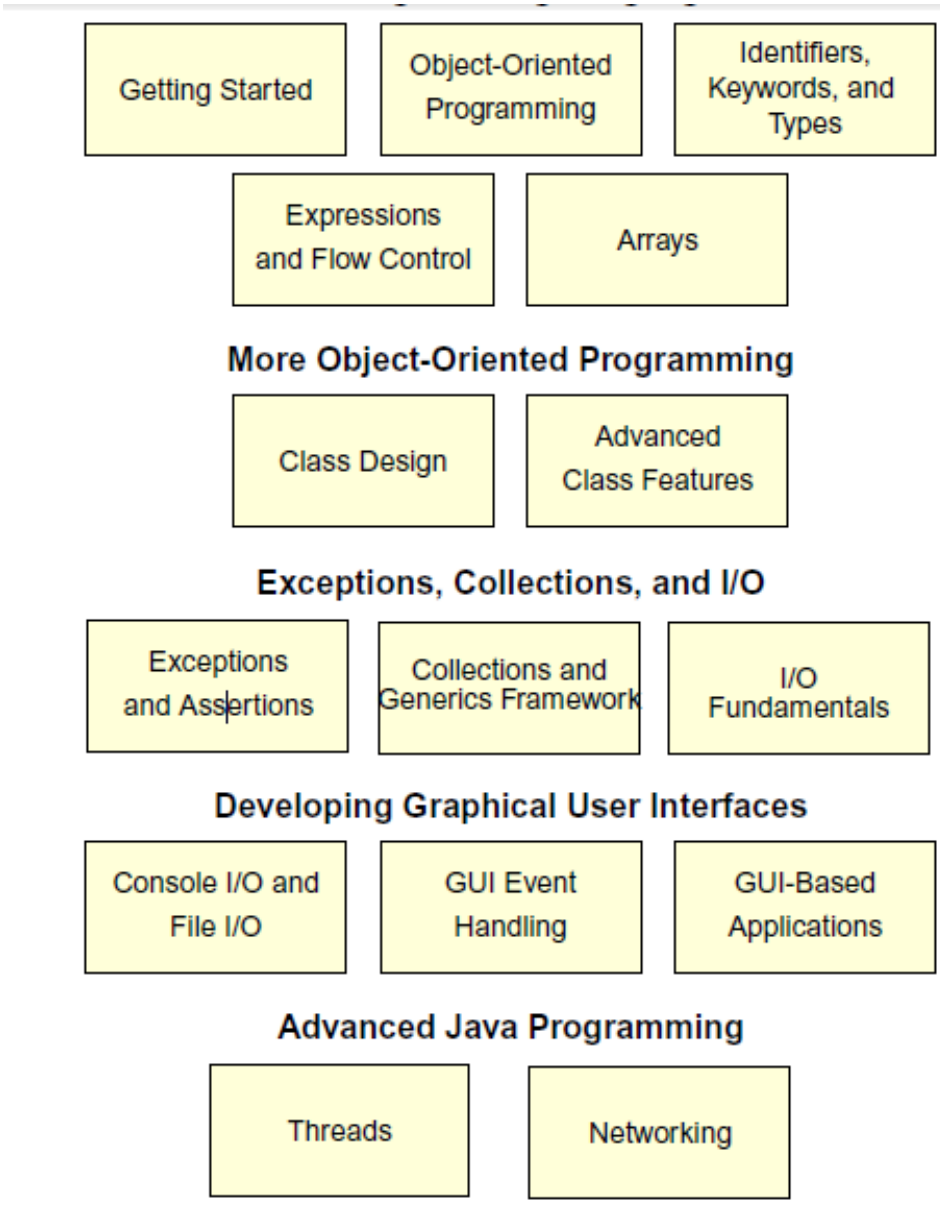

Fig 1: Java programming Basics

4. The Java Runtime Environment

The Java application environment performs as follows:

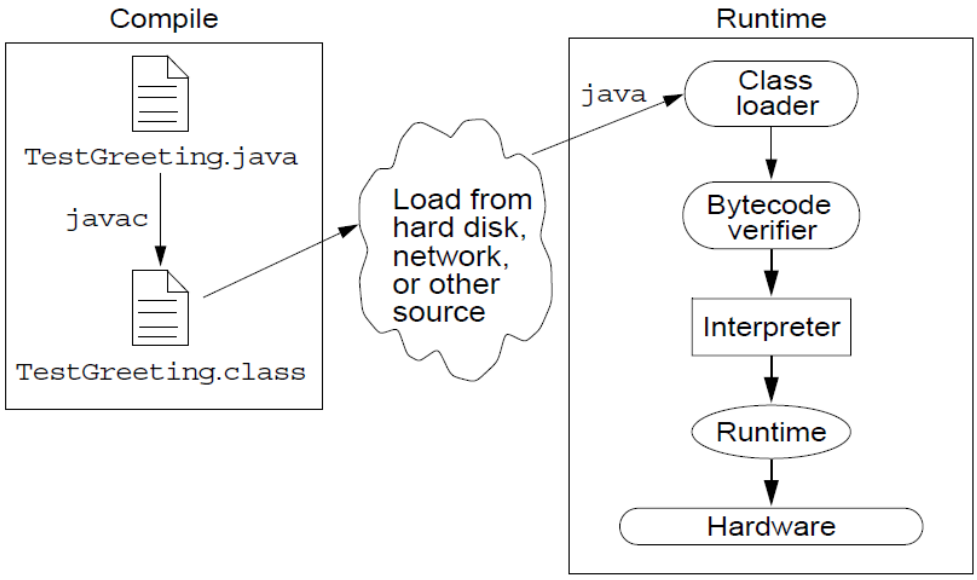

Fig 2: Java Runtime Environment (JRE)

In some Java technology runtime environments, apportion of the verified bytecode is compiled to native machine code and executed directly on the hardware platform.
What is the difference to use that?

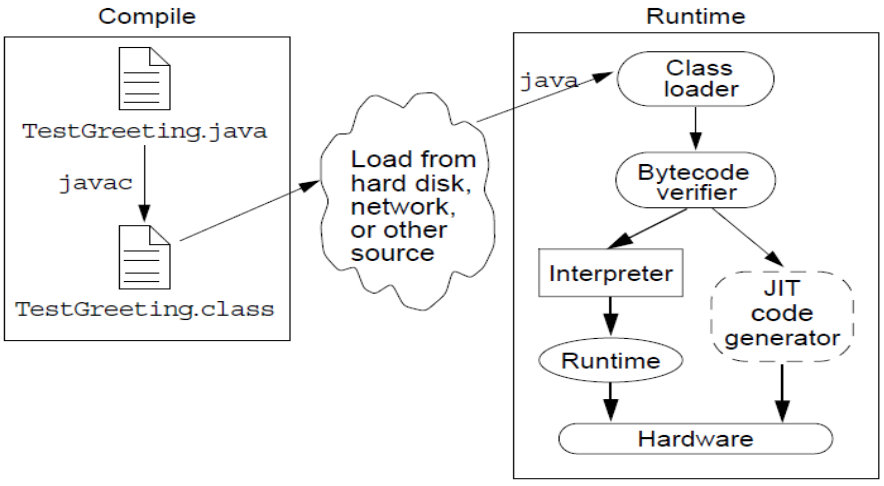

Fig 2: Java Runtime Environment (JRE) with Just in Time (JIT)

A Simple Java Application

The TestGreeting.java Application

```
1   //
2   // Sample "Hello World" application
3   //
4   public class TestGreeting{
5     public static void main (String[] args) {
6       Greeting hello = new Greeting();
7       hello.greet();
8     }
9   }
```

The Greeting.java Class

```
1   public class Greeting {
2     public void greet() {
3       System.out.println("hi");
4     }
5   }
```

Fig 3: Simple Class (Greeting Class)

Questions and Exercises: Getting Started

Questions

Question 1: When you compile a program written in the Java programming language, the compiler converts the human-readable source file into platform-independent code that a Java Virtual Machine can understand. What is this platform-independent code called?

Question 2: Which of the following is *not* a valid comment:

a. `/** comment */`
b. `/* comment */`
c. `/* comment`
d. `// comment`

Question 3: What is the first thing you should check if you see the following error at runtime:

```
Exception in thread "main" java.lang.NoClassDefFoundError:
HelloWorldApp.java.
```

Question 4: What is the correct signature of the `main` method?

Question 5: When declaring the `main` method, which modifier must come first, `public` or `static`?

Question 6: What parameters does the `main` method define?

Exercises

Exercise 1: Change the `HelloWorldApp.java` program so that it displays `Hola Mundo!` instead of `Hello World!`.

Exercise 2: You can find a slightly modified version of `HelloWorldApp` here: `HelloWorldApp2.java`

```
class HelloWorldApp2 {
    public static void main(String[] args) {
        System.out.println("Hello World!); // Display the string.
    }}
```

The program has an error. Fix the error so that the program successfully compiles and runs. What was the error?


جامعة بابل
UNIVERSITY OF BABYLON