Course: Object Based Modeling
Code: CS-33105
Branch: MCA-3

Lecture - 1

Faculty: Dr. J Sathish Kumar (JSK) – Coordinator

Department of Computer Science and Engineering

Motilal Nehru National Institute of Technology Allahabad,
Prayagraj-211004

Stay Safe!!!

COVID-19 Dashboard
as on : 10 August 2020, 08:00 IST (GMT+5:30)

Total Cases
22,15,074
62064 ↑

Active
(28.66%)
6,34,945
6198 ↑

Discharged
(69.33%)
15,35,743
54859 ↑

Deaths
(2.00%)
44,386
1007 ↑

Need Your Co-operation!!

Curriculum

Course Outline (To be covered in 30 lectures)

1. *Core Java*: Introduction to Object Oriented Software development through Java. Classes and Objects. (6)
2. Inheritance, Polymorphism, Nested classes and interfaces, Exceptions, Strings, Packages, The I/O Package.(8)
3. *Advanced Java*: Event Handling, AWT, Swing, Applets, Multi-Threading, Generic, The collection frameworks.(8)
4. Networking, Java Server Pages (JSP), Java Servlet, Enterprise Java Beans (EJB), Java Messaging Service (JMS), Java Database connectivity (JDBC).(6)

Text Books

1. Kathy Sierra and Bert Bates, “Head First Java”, 2nd edition, O’Reilly
2. Herbert Schildt , “Java : The Complete Reference”, 9th edition, Oracle Press
3. Cay S. Horstmann and Gary Cornell , “Core Java Volume I & II”, 10th edition, Prentice-Hall
4. Tony Gaddis, “Starting Out with Java: From Control Structures through Objects”, 6th edition, Pearson
5. David Flanagan, “Java in a Nutshell”, 5th edition, O’Reilly

Legacy of Computer language

- Computer language innovation and development occurs for two fundamental reasons:
  - To adapt to changing environments and uses
  - To implement refinements and improvements in the art of programming
- FORTRAN could be used to write fairly efficient programs for scientific applications, it was not very good for system code.
- BASIC was easy to learn, it wasn’t very powerful, and its lack of structure made its usefulness questionable for large programs.
- Assembly language can be used to produce highly efficient programs, but it is not easy to learn or use effectively. Further, debugging assembly code can be quite difficult.

Legacy of Computer language

- While languages like Pascal are structured, they were not designed for efficiency, and failed to include certain features necessary to make them applicable to a wide range of programs.
- The creation of C is considered by many to have marked the beginning of the modern age of computer languages.
  - It successfully synthesized the conflicting attributes that had so troubled earlier languages.
  - The result was a powerful, efficient, structured language that was relatively easy to learn.
- Throughout the history of programming, the increasing complexity of programs has driven the need for better ways to manage that complexity.

Legacy of Computer language

- The first widespread language was, of course, FORTRAN. While FORTRAN was an impressive first step, it is hardly a language that encourages clear and easy-to-understand programs.
- The 1960s gave birth to *structured programming*.
  - This is the method of programming championed by languages such as C.
- However, even with structured programming methods, once a project reaches a certain size, its complexity exceeds what a programmer can manage.
- By the early 1980s, many projects were pushing the structured approach past its limits.
- To solve this problem, a new way to program was invented, called *object-oriented programming (OOP)*.
- OOP is a programming methodology that helps organize complex programs through the use of inheritance, encapsulation, and polymorphism.

Two Paradigms

- All computer programs consist of two elements: code and data.
- The first way is called the *process-oriented model*.
  - This approach characterizes a program as a series of linear steps (that is, code).
  - The process-oriented model can be thought of as *code acting on data*.
  - Procedural languages such as C employ this model to considerable success.
  - Problems with this approach appear as programs grow larger and more complex.
- To manage increasing complexity, the second approach, called *object-oriented programming*, was conceived.
  - Object-oriented programming organizes a program around its data (that is, objects) and a set of well-defined interfaces to that data.
  - An object-oriented program can be characterized as *data controlling access to code*.

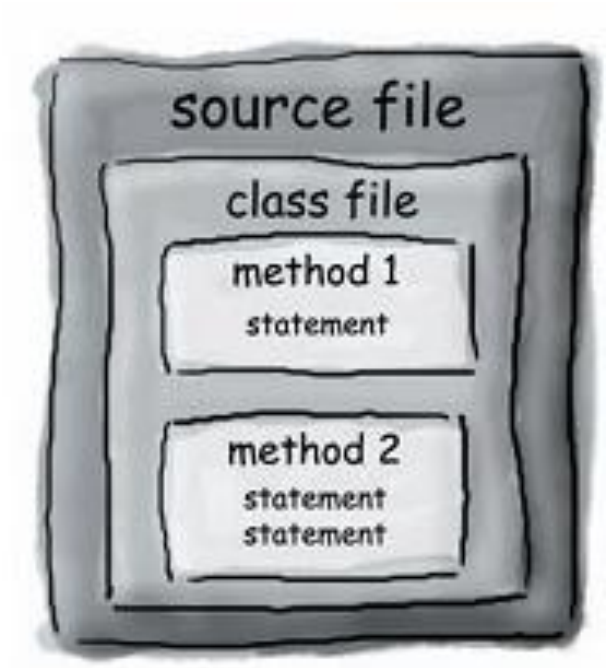
Legacy of Computer language

- Stroustrup initially called the new language “C with Classes.” However, in 1983, the name was changed to C++.
- C++ extends C by adding object-oriented features.
- Because C++ is built on the foundation of C, it includes all of C’s features, attributes, and benefits.
- This is a crucial reason for the success of C++ as a language.
- The invention of C++ was not an attempt to create a completely new programming language. Instead, it was an enhancement to an already highly successful one.

Legacy of Computer language

- Java was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan at Sun Microsystems, Inc. in 1991.
  - It took 18 months to develop the first working version.
  - This language was initially called “Oak,” but was renamed “Java” in 1995.
- Java enhanced and refined the object-oriented paradigm used by C++, added integrated support for multithreading, and provided a library that simplified Internet access.
- While it is true that Java was influenced by C++, it is not an enhanced version of C++.
- Java features
  - Simple, Secure, Portable Object-oriented, Robust, Multithreaded, Architecture-neutral, Interpreted, High performance, Distributed and Dynamic

An Introduction to Java



Put a class in a source file.

Put methods in a class.

Put statements in a method.

What goes in a source file?

```
public class Dog {

}
```

class

What goes in a class?

```
public class Dog {
   void bark() {

   }
}
```

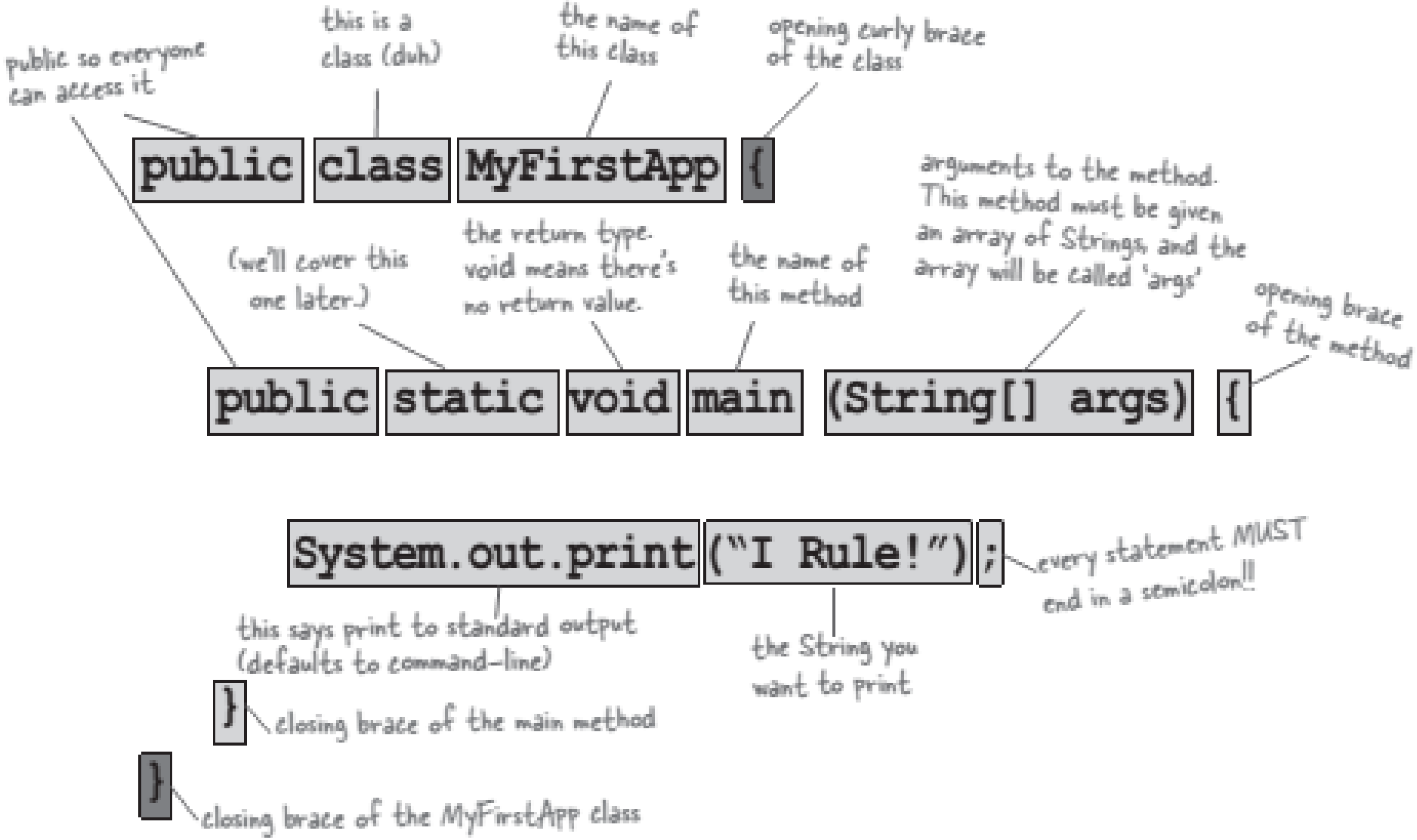
method

What goes in a method?

```
public class Dog {
   void bark() {
    statement1;
    statement2;
   }
}
```

statements

Anatomy of a class



Compile and Execute

1 Save

```
MyFirstApp.java
```

2 Compile

```
javac MyFirstApp.java
```

3 Run

File Edit Window Help Scream

```
%java MyFirstApp
I Rule!
The World
```

Syntax

- Each statement must end in a semicolon.
  - x = x + 1**;**
- A single-line comment begins with two forward slashes.
  - x = 22;
    // this line disturbs me
- Most white space doesn't matter.
  - x = 3 ;
- Variables are declared with a **name** and a **type**
  - int weight;
    //*type:* int, *name:* weight
- Classes and methods must be defined within a pair of curly braces.
  - public void go()
    {
    // amazing code here
    }

Statements

Declarations, assignments, method calls, etc.

```
int x = 3;
String name = "Dirk";
x = x * 17;
System.out.print("x is " + x);
double d = Math.random();
// this is a comment
```

Branching: *if/else* tests

```
if (x == 10) {
   System.out.print("x must be 10");
} else {
   System.out.print("x isn't 10");
}
if ((x < 3) & (name.equals("Dirk"))) {
   System.out.println("Gently");
}
System.out.print("this line runs no matter what");
```

Loops: *for* and *while*

```
while (x > 12) {
    x = x -1;
}

for (int x = 0; x < 10; x = x + 1) {
    System.out.print("x is now " + x);
}
```

Class Exercise #1

```
public class Loopy {
   public static void main (String[] args) {
     int x = 1;
     System.out.println("Before the Loop");
     while (x < 4) {
       System.out.println("In the loop");
       System.out.println("Value of x is " + x);
         x = x + 1;
     }
     System.out.println("This is after the loop");
   }
}
```

```
% java Loopy
Before the Loop
In the loop
Value of x is 1
In the loop
Value of x is 2
In the loop
Value of x is 3
This is after the loop
```

this is the output

Class Exercise #2

```
class IfTest {
  public static void main (String[] args) {
    int x = 3;
    if (x == 3) {
      System.out.println("x must be 3");
    }
    System.out.println("This runs no matter what");
  }
}
```

```
% java IfTest
x must be 3
This runs no matter what
```

code output

Class Exercise #3

Given the output:

```
% java DooBee
DooBeeDooBeeDo
```

Fill in the missing code:

```
public class DooBee {
  public static void main (String[] args) {
    int x = 1;
    while (x < _____ ) {
      System.out._________("Doo");
      System.out._________("Bee");
      x = x + 1;
    }
    if (x == _______  ) {
      System.out.print("Do");
    }
  }
}
```

Class Exercise #4

Given the output:

```
% java Shuffle1
a-b c-d
```

```
if (x == 1) {
   System.out.print("d");
   x = x - 1;
}
```

```
if (x > 2) {
   System.out.print("a");
}
```

```
int x = 3;
```

```
if (x == 2) {
   System.out.print("b c");
}
```

```
x = x - 1;
System.out.print("-");
```

```
while (x > 0) {
```

```
class Shuffle1 {
  public static void main(String [] args) {
```

Class Exercise #5

A

```
class Exercise1b {
  public static void main(String [] args) {
    int x = 1;
    while ( x < 10 ) {
      if ( x > 3) {
        System.out.println("big x");
      }
    }
  }
}
```

B

```
public static void main(String [] args) {
    int x = 5;
    while ( x > 1 ) {
      x = x - 1;
      if ( x < 3) {
        System.out.println("small x");
      }
    }
}
```

C

```
class Exercise1b {
    int x = 5;
    while ( x > 1 ) {
      x = x - 1;
      if ( x < 3) {
        System.out.println("small x");
      }
    }
}
```

BE the compiler

Identify the correct program!!

Object-Oriented Programming Features in Java

- Abstraction
- Encapsulation
- Inheritance
- Polymorphism