Object-Oriented Programming

BITS Pilani
Hyderabad Campus

Getting Started

Announcement

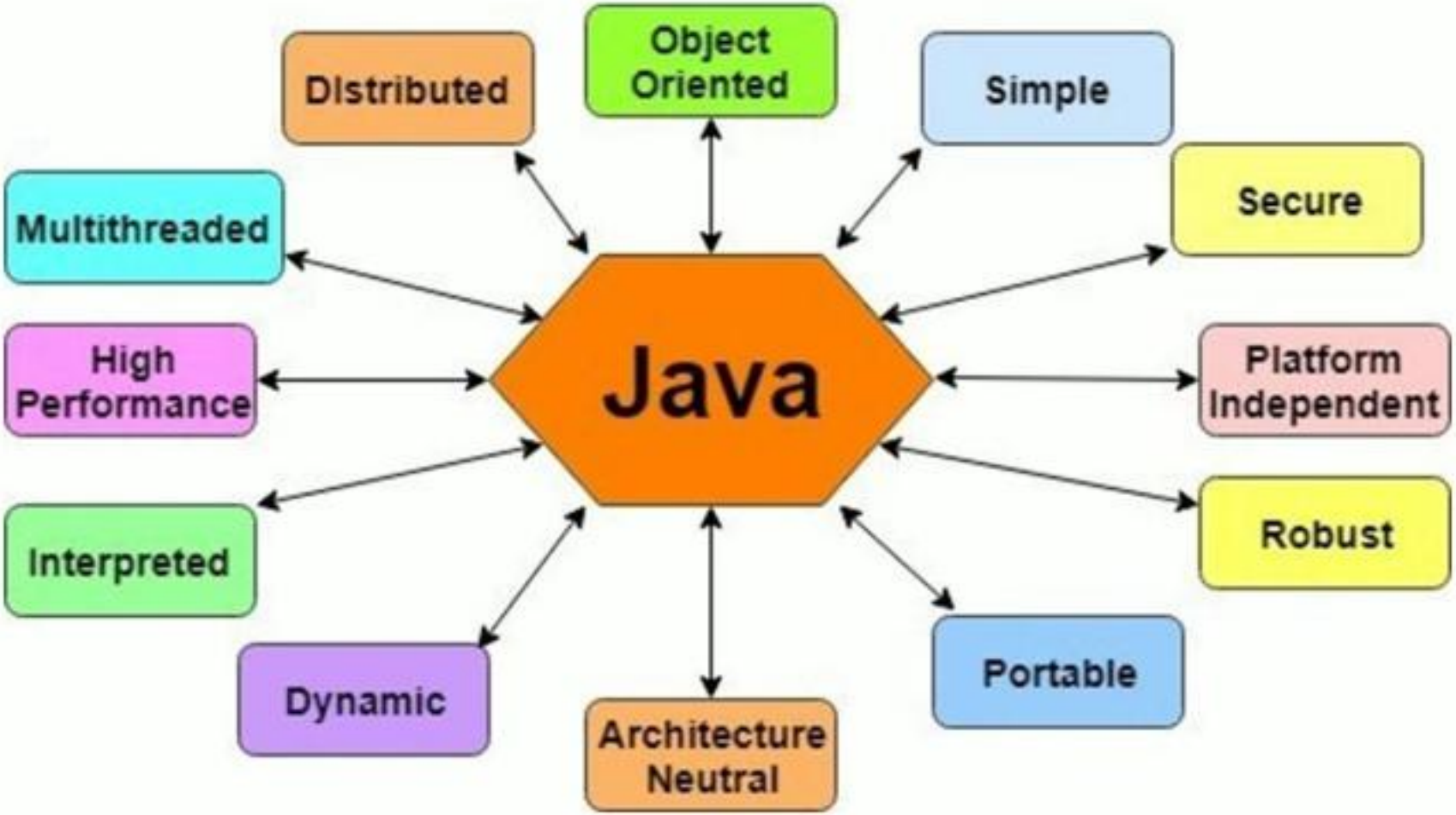
Note:

These slides are shared with the students of BITS-Pilani Hyderabad Campus Students enrolled in Object-Oriented Programming (CSF213) Course of Semester-II, 2019-2020 academic session, purely for their academic reference and study. In case it is shared, distributed or sold to anyone outside (not enrolled in the stated course) without approval of the Instructor In-Charge will be considered as breach of trust and academic integrity.

Best Wishes,

Dr. Subhrakanta Panda, *Ph.D.*
Assistant Professor, CSIS, BPHC.
IC, OOP (CSF213) Course,
Semester-2, 2019-2020.

Revisit: Features of Java



Getting Started with Java

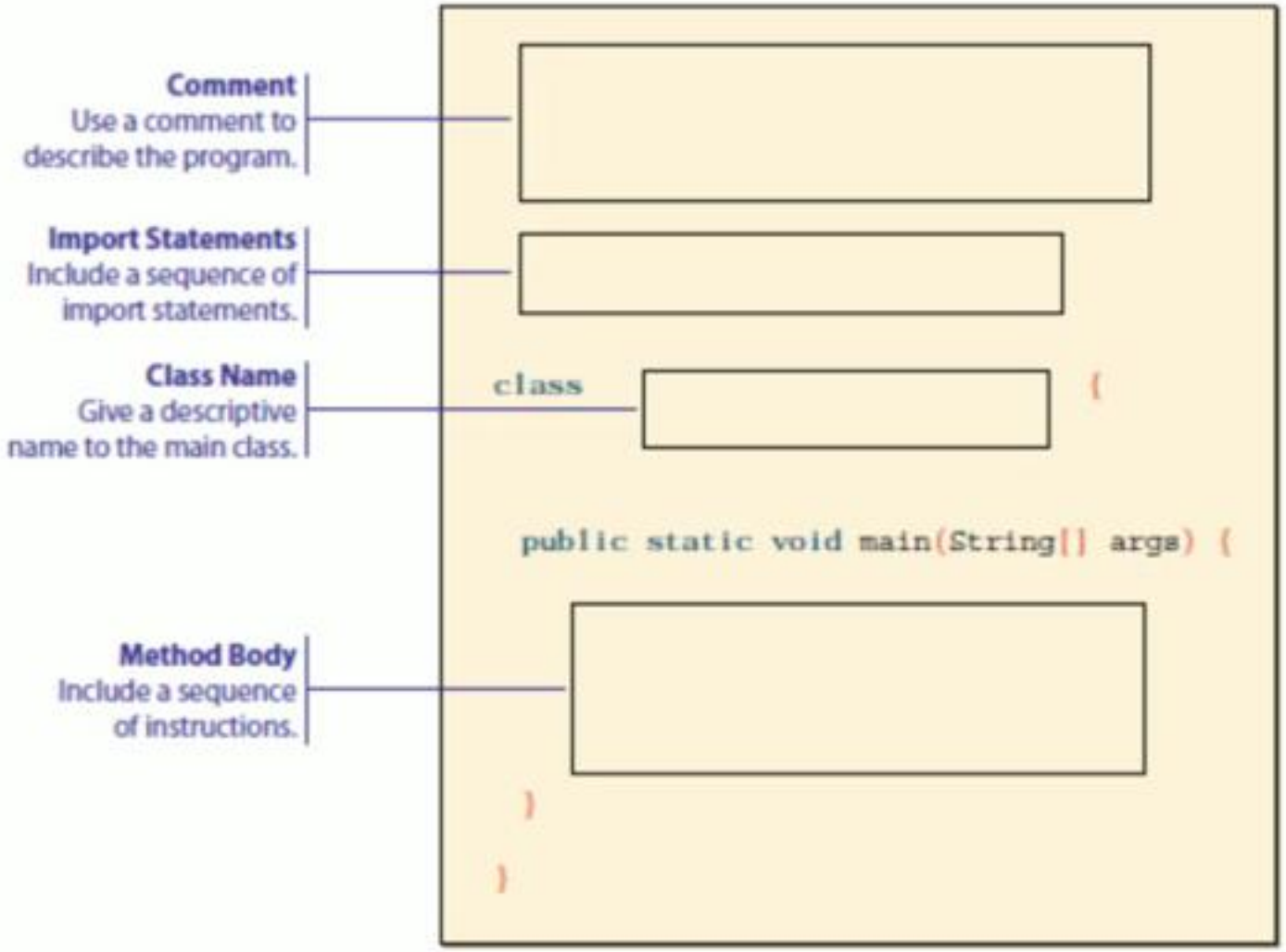
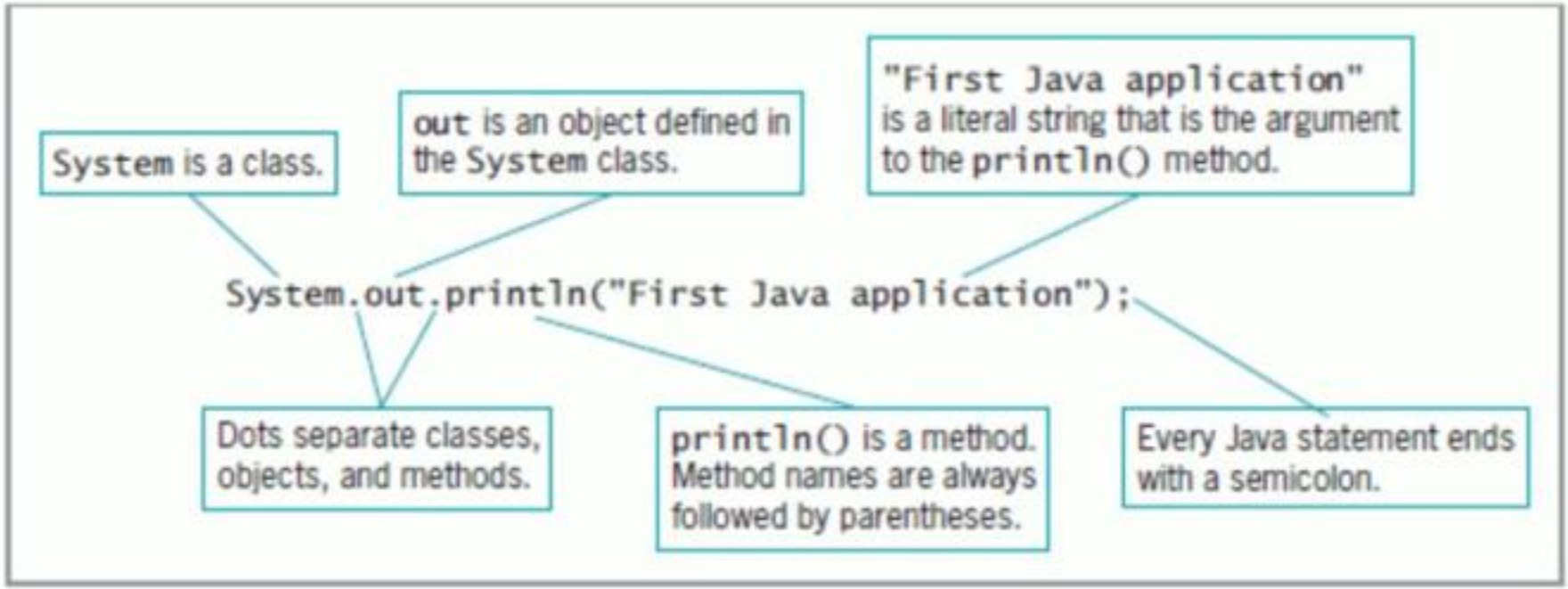


Figure 2.9 A program template for simple Java programs.

Getting Started with Java

innovate achieve lead

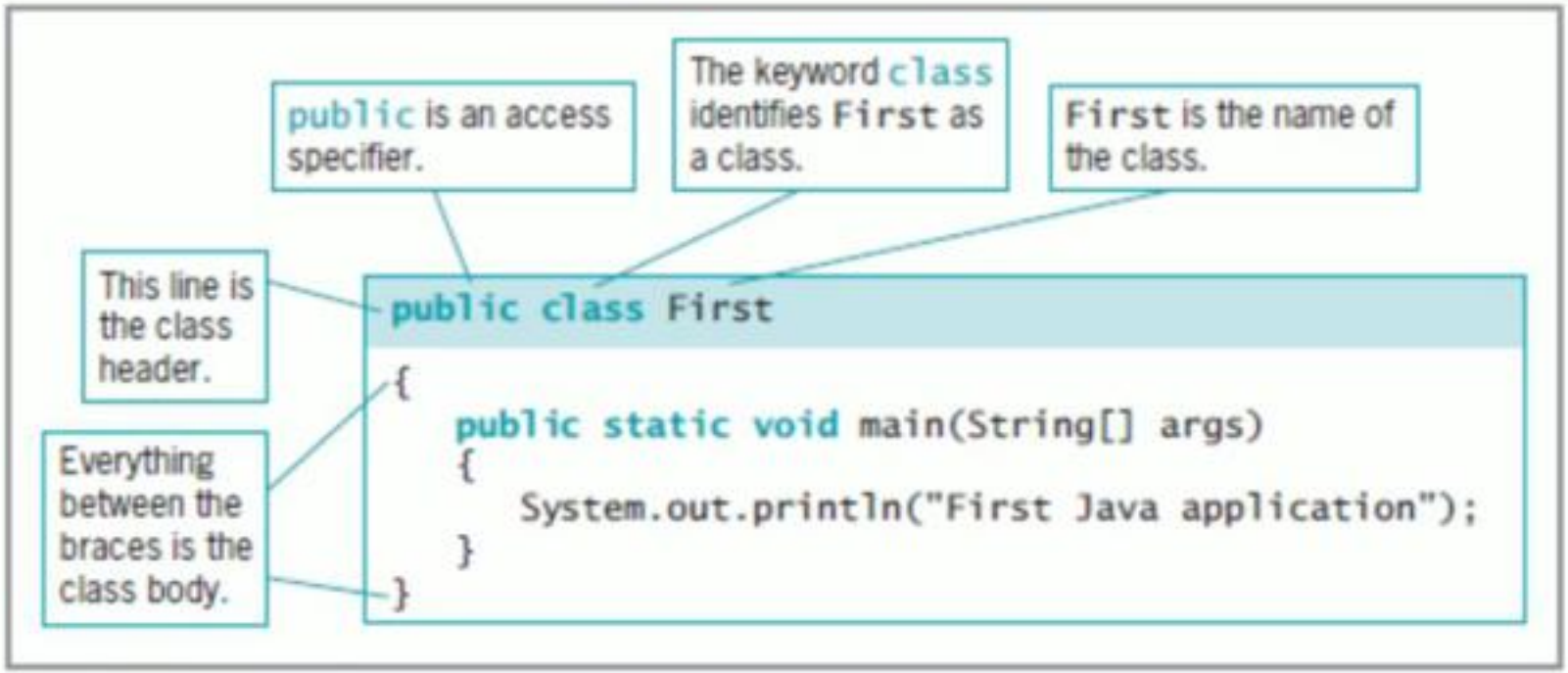
```
import java.lang.*;
import java.io.*;
import java.util.*;
import java.applet.*;
import java.awt.*;
1.class First{
2.    public static void main(String args[])
3.{
4.     System.out.println("First Java application");
5.    }
6.}
```



Getting Started with Java

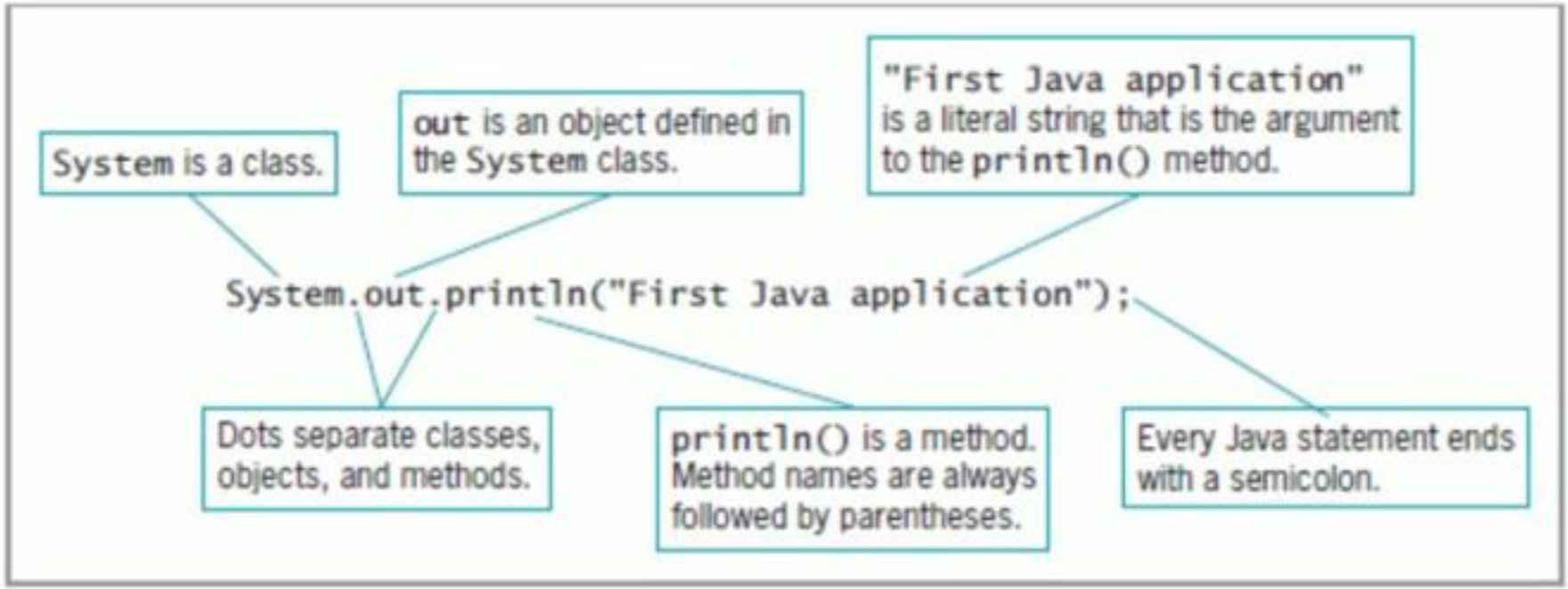
innovate achieve lead

```
import java.lang.*;
import java.io.*;
import java.util.*;
import java.applet.*;
import java.awt.*;
1.class First{
2.    public static void main(String args[])
3.{
4.     System.out.println("First Java application");
5.    }
6.}
```



Getting Started with Java

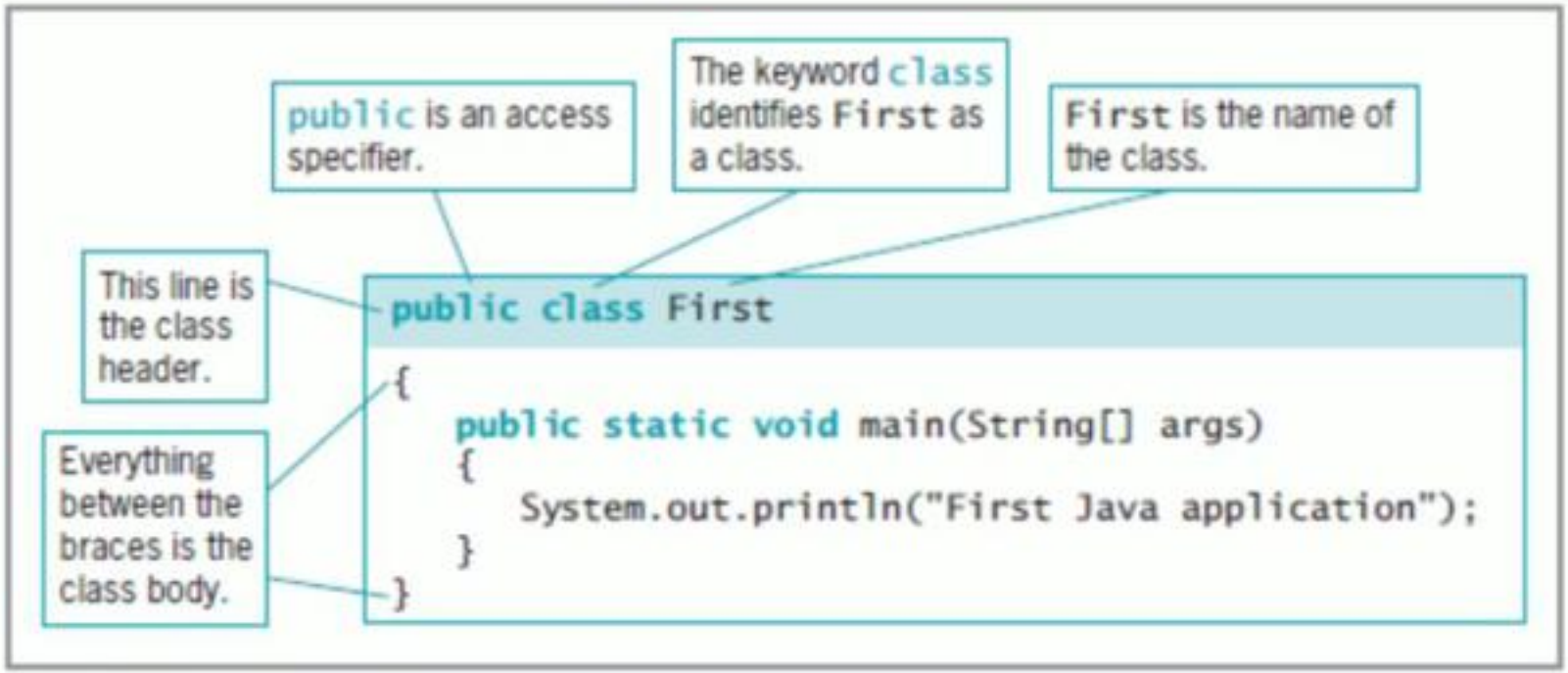

```
import java.lang.*;
import java.io.*;
import java.util.*;
import java.applet.*;
import java.awt.*;
1.class First{
2.    public static void main(String args[])
3.{
4.     System.out.println("First Java application");
5.    }
6.}
```



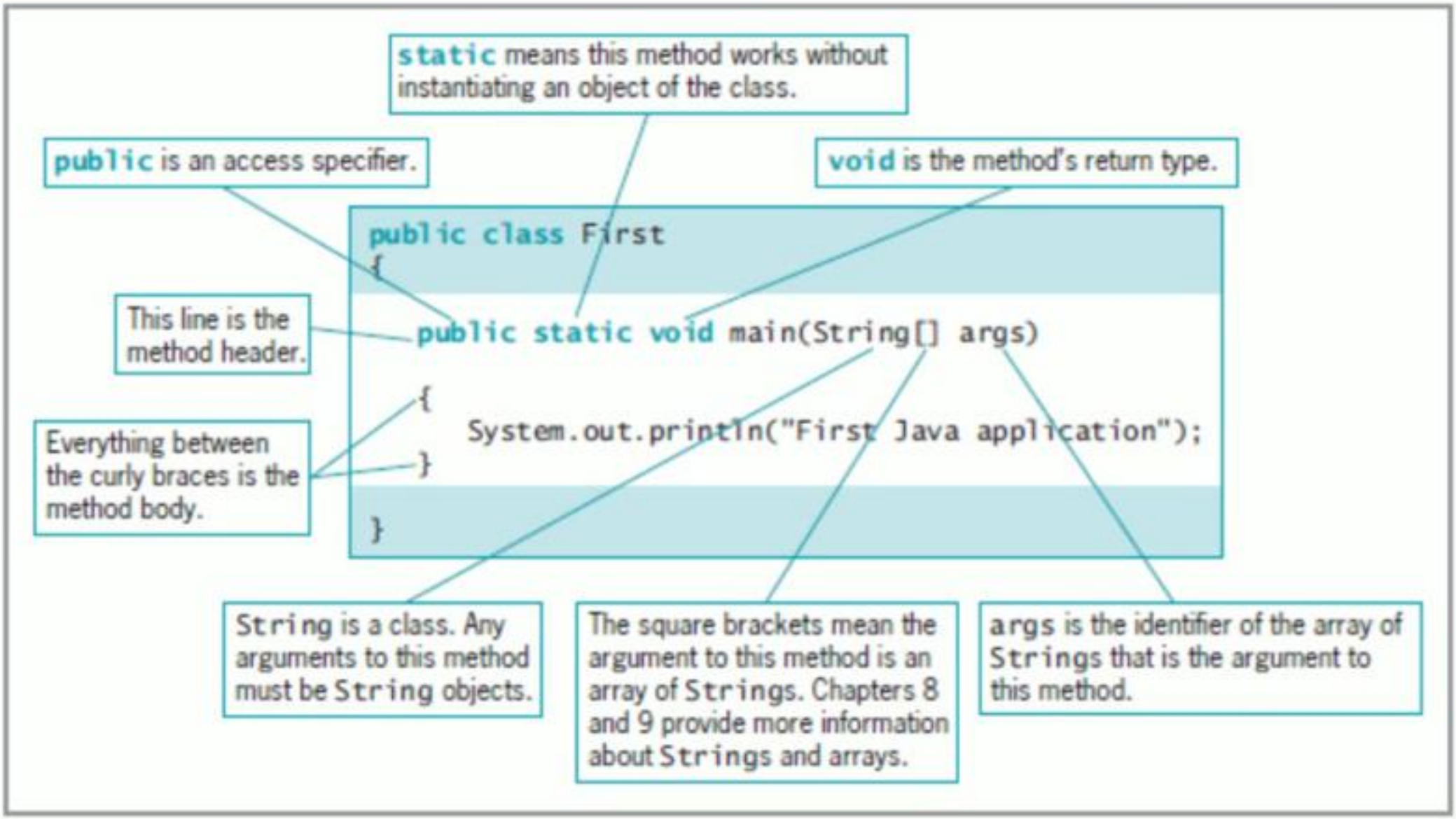
Getting Started with Java

innovate achieve lead

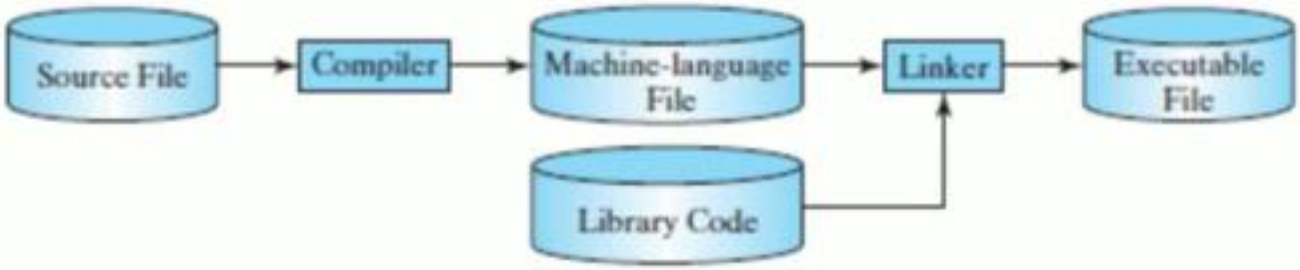
```
import java.lang.*;
import java.io.*;
import java.util.*;
import java.applet.*;
import java.awt.*;
1.class First{
2.    public static void main(String args[])
3.{
4.     System.out.println("First Java application");
5.    }
6.}
```

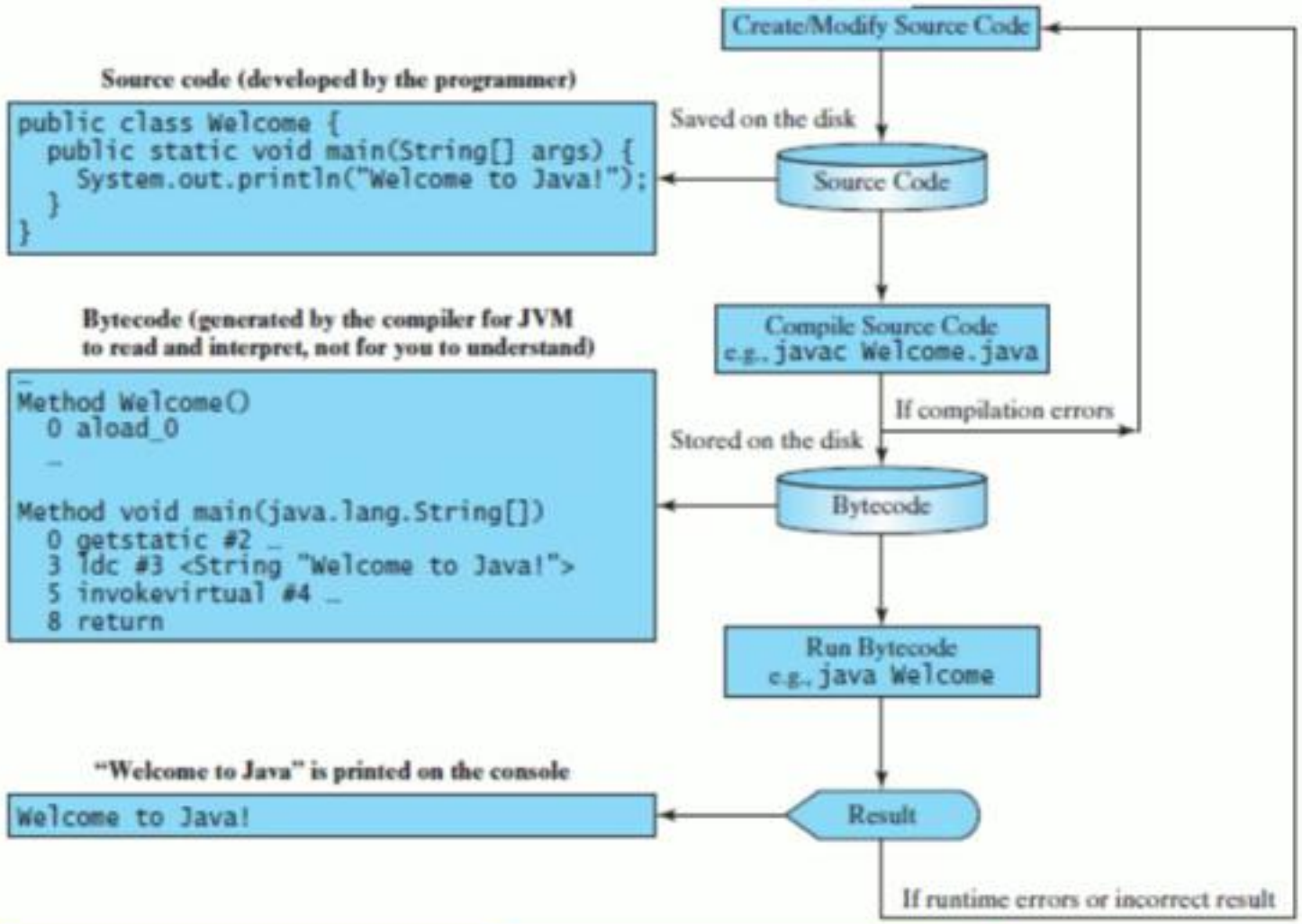


Getting Started with Java

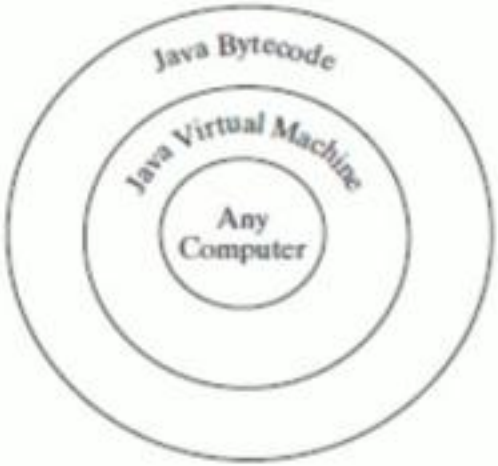


Creating, Compiling, Executing

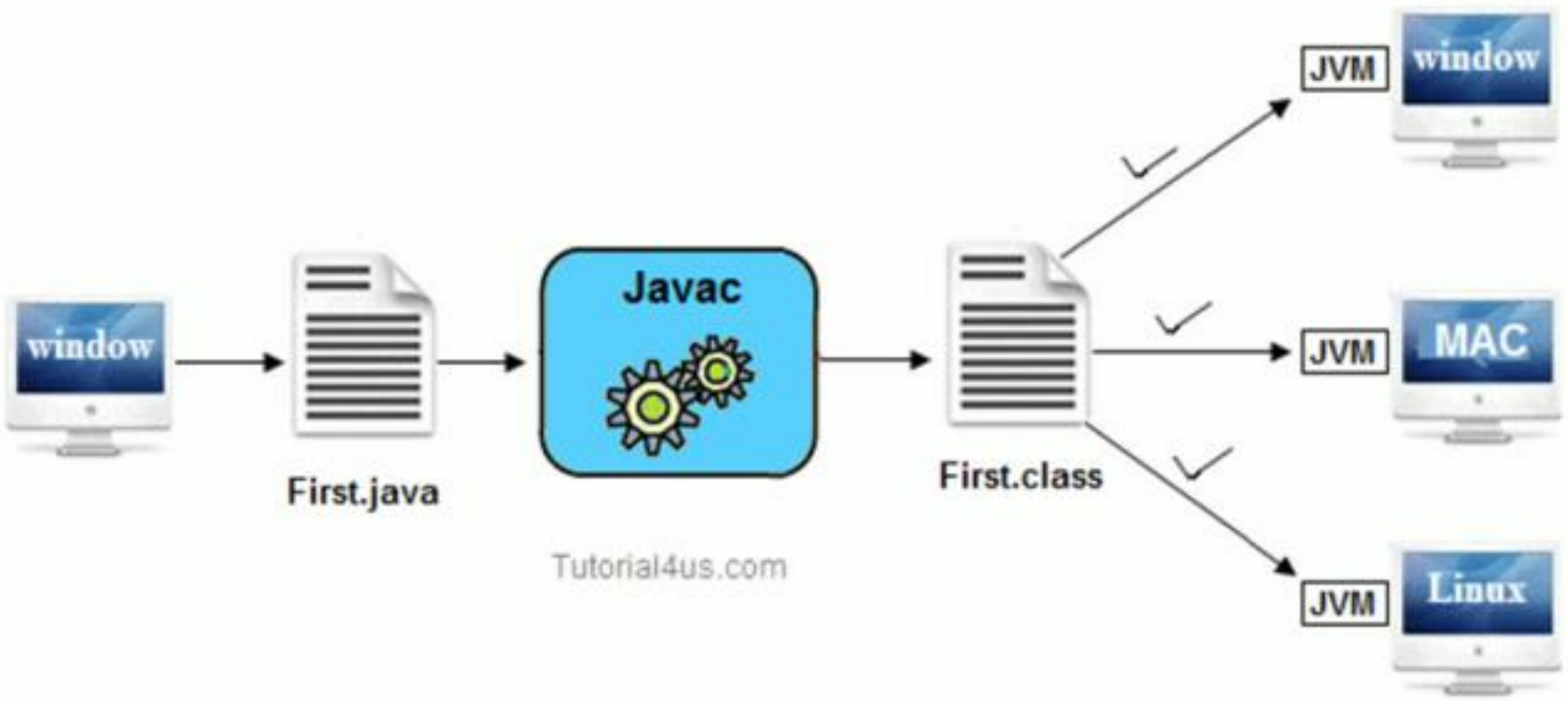




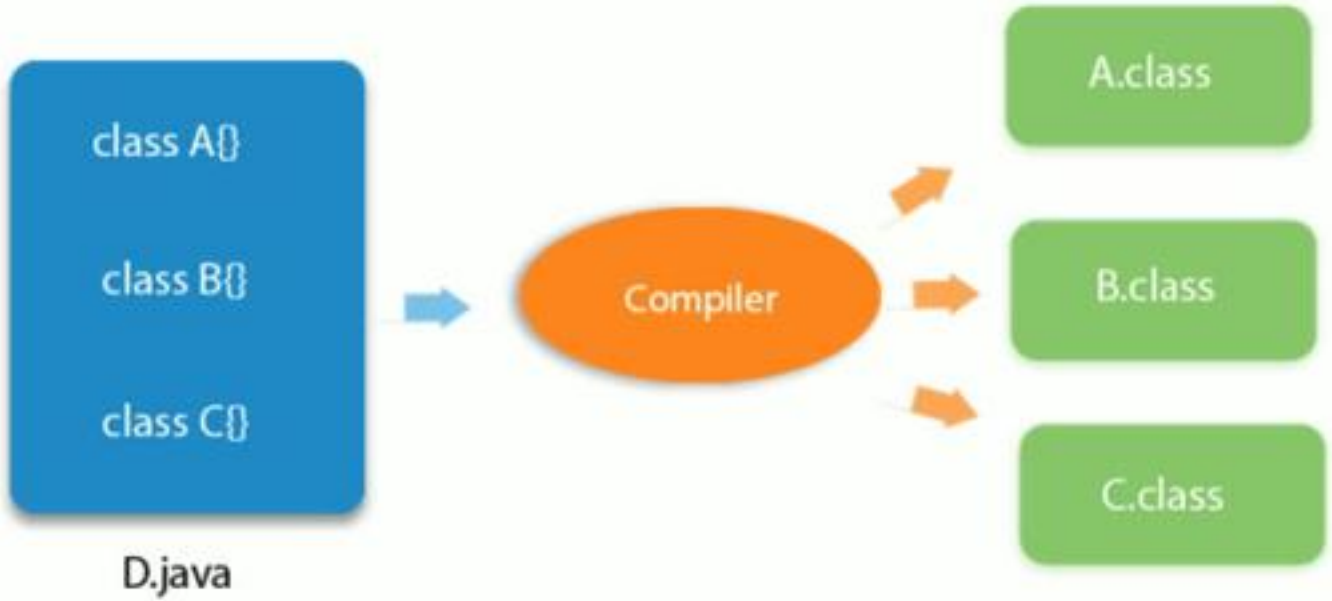
Creating, Compiling, Executing



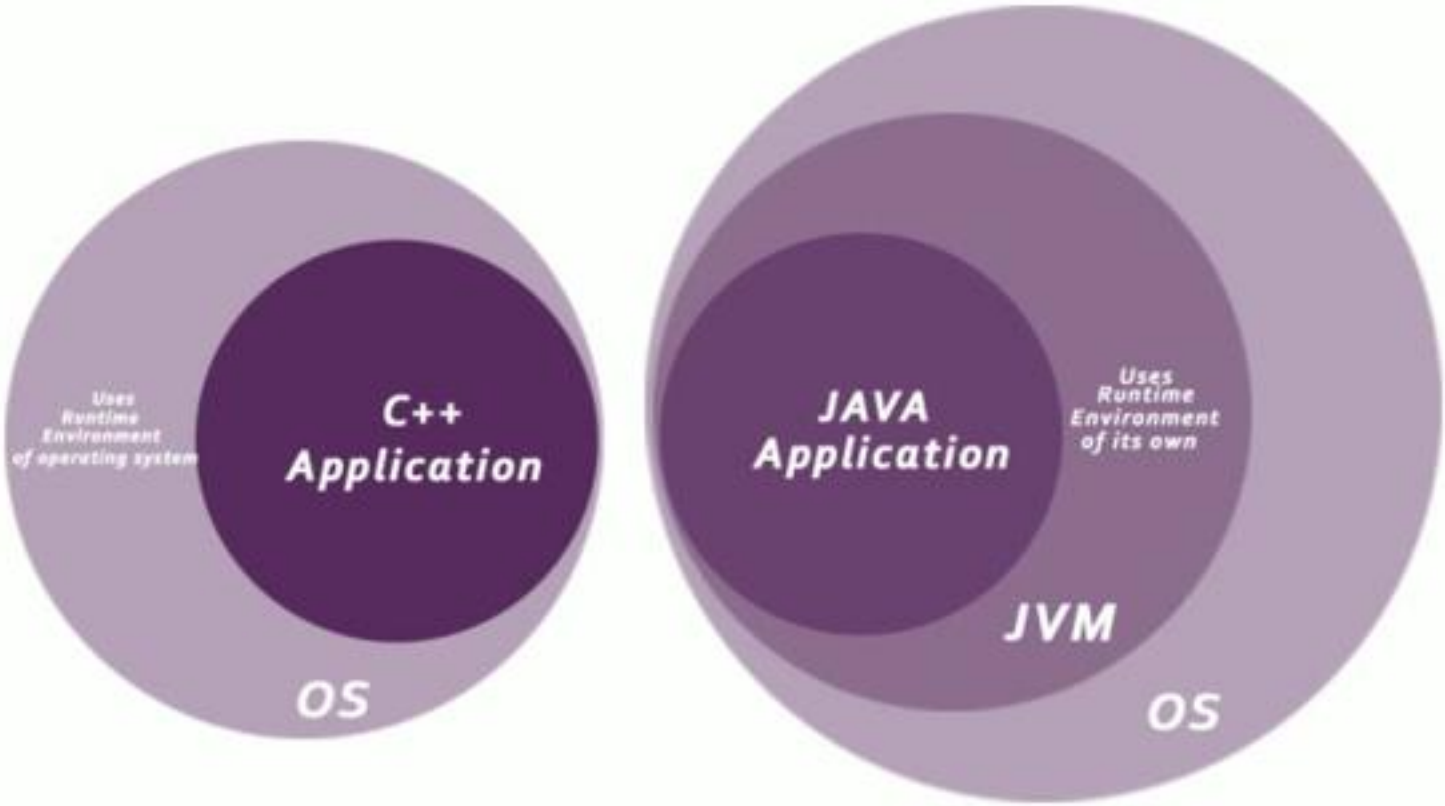
Getting Started with Java



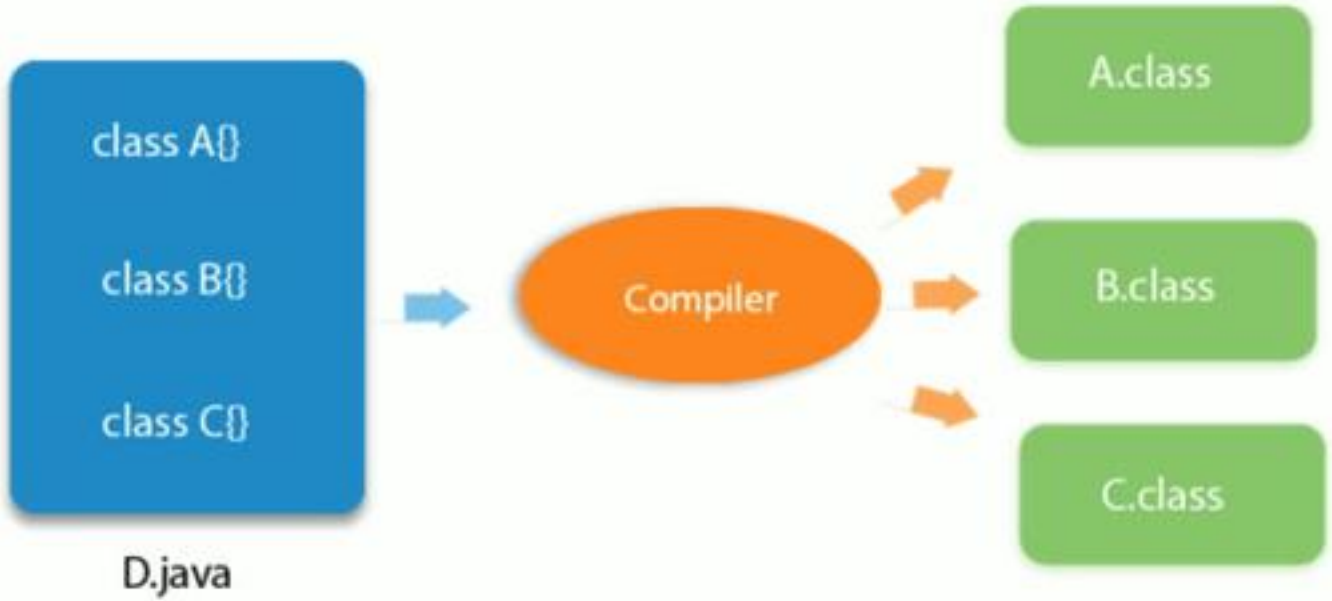
Getting Started with Java



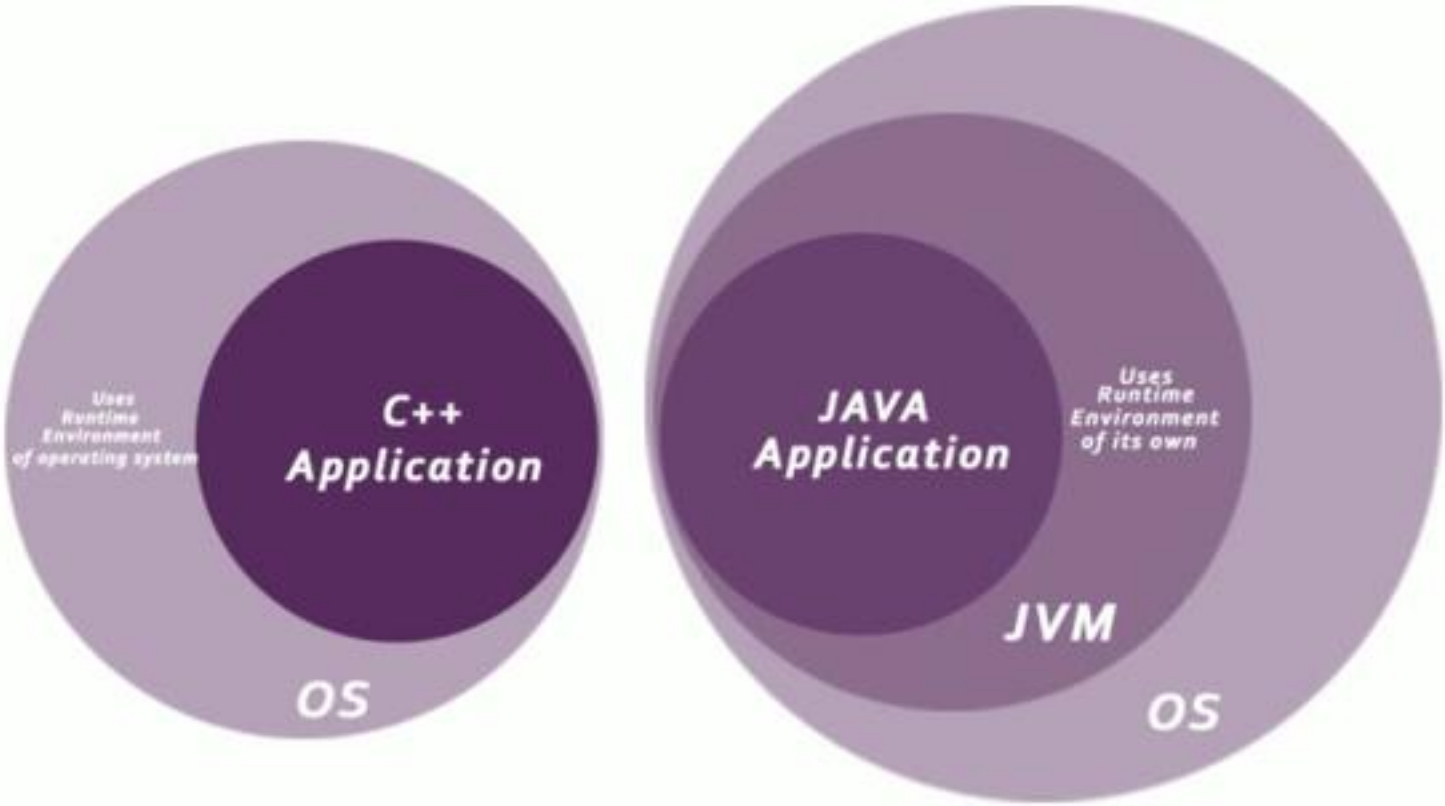
Getting Started with Java



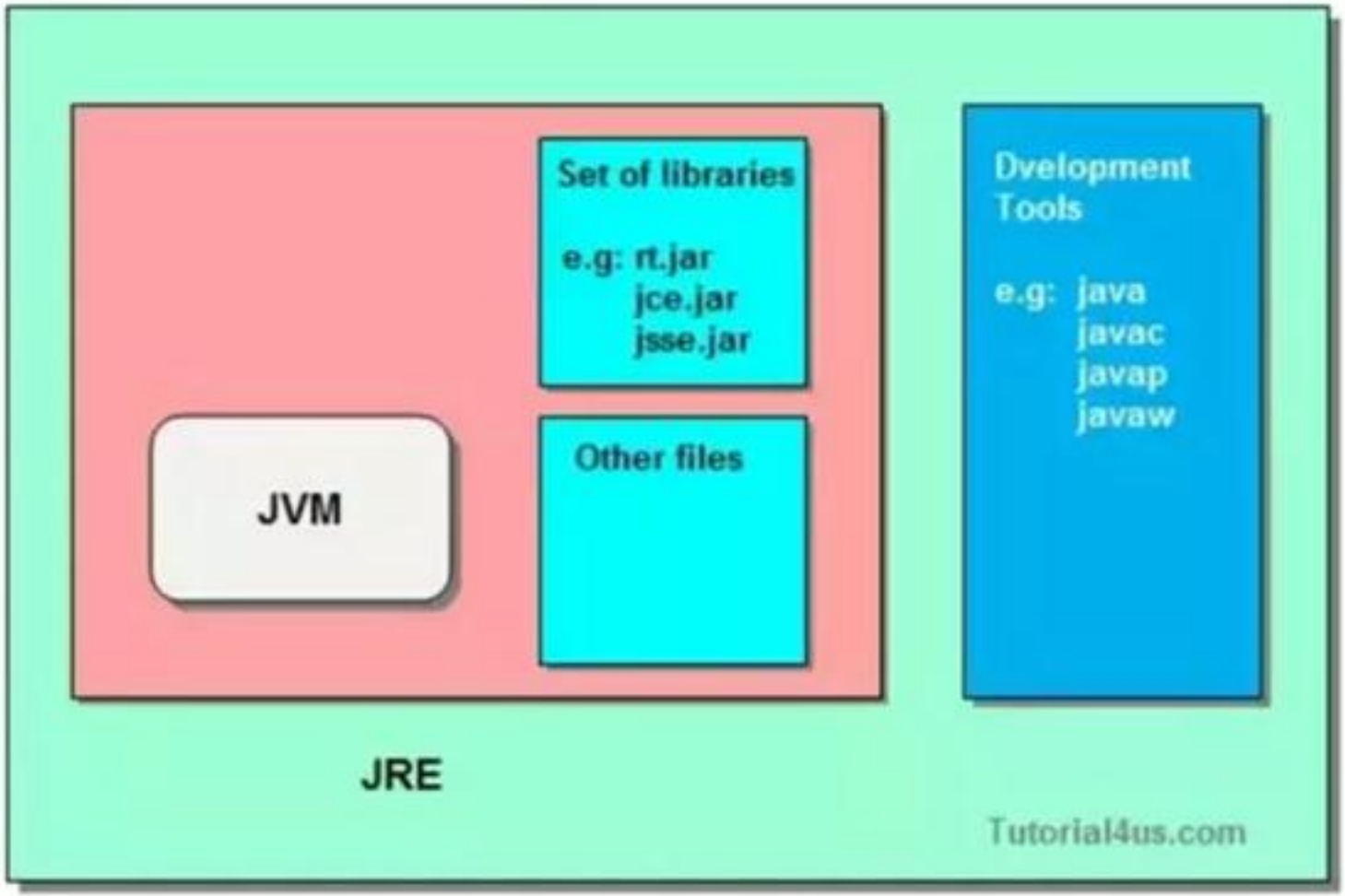
Getting Started with Java



Getting Started with Java



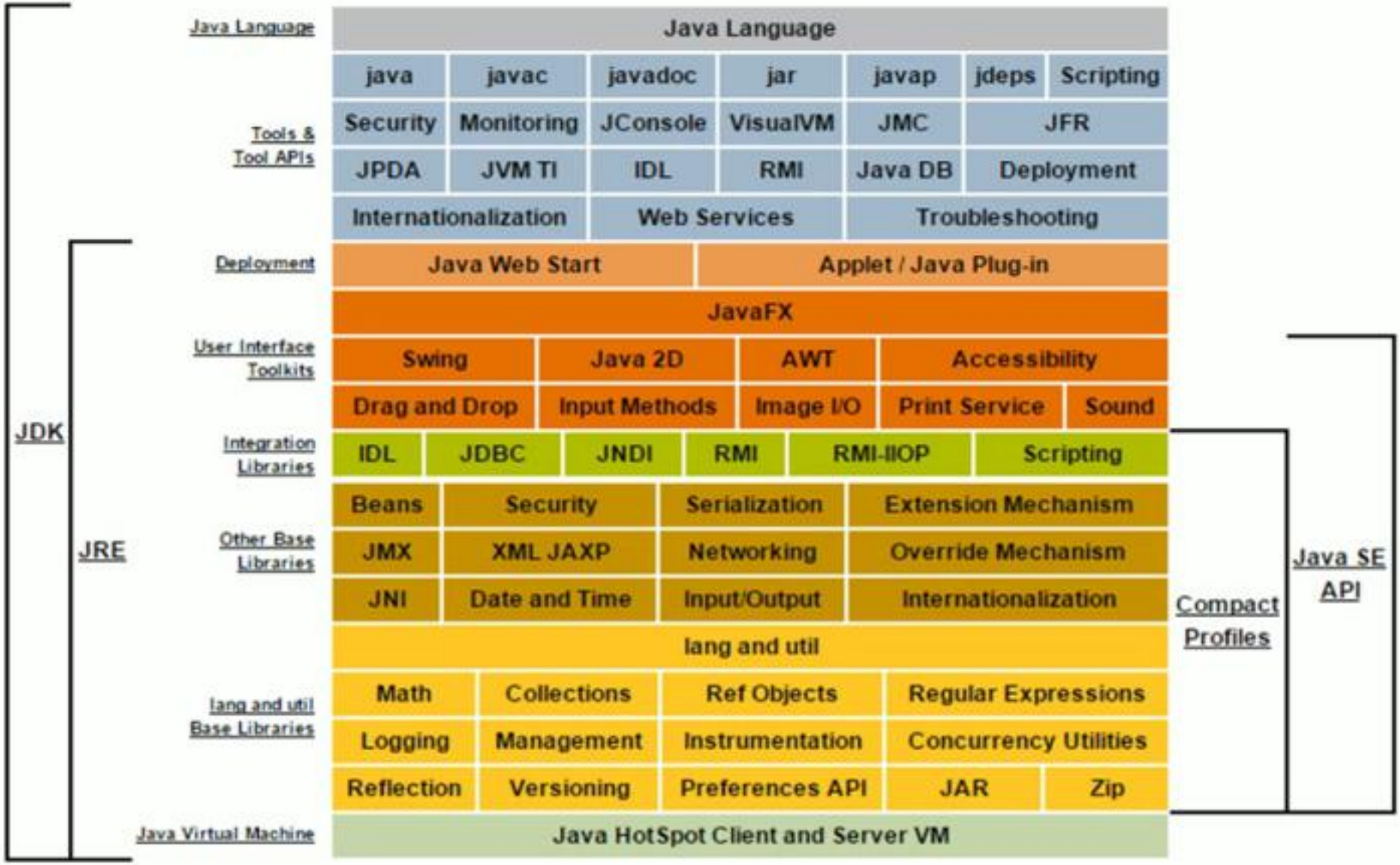
Getting Started with Java



Getting Started with Java

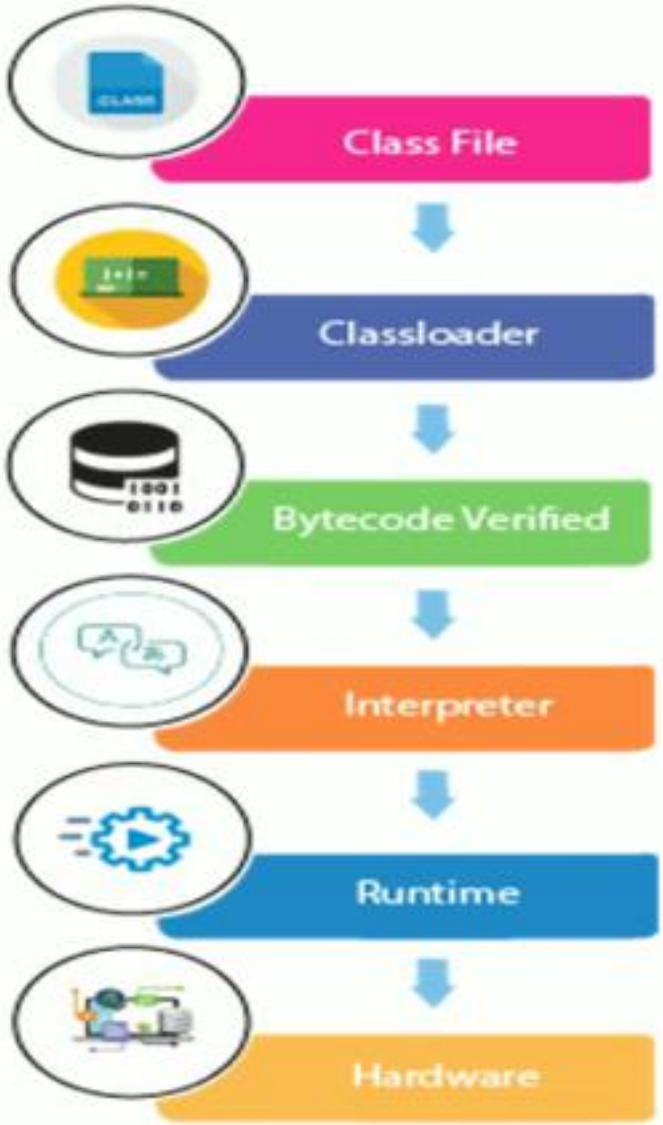
innovate achieve lead

Description of Java Conceptual Diagram

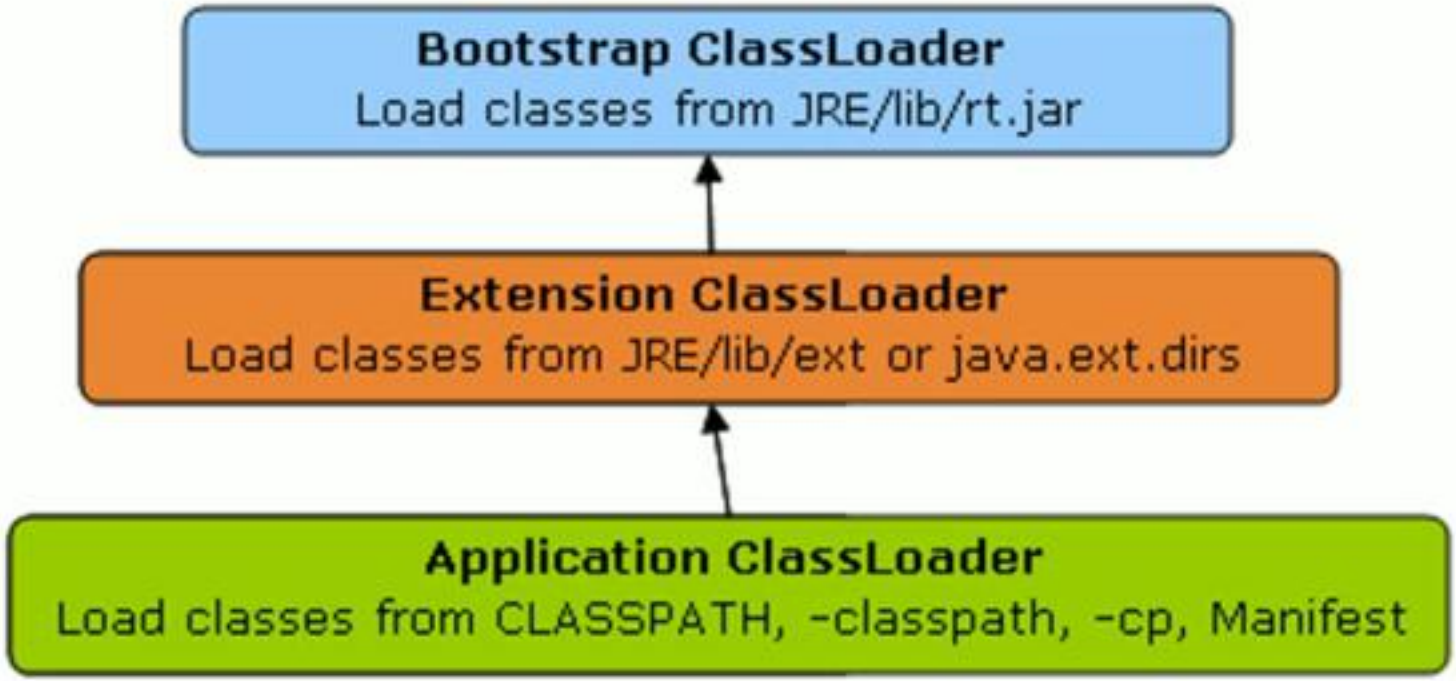


Getting Started with Java

At runtime, following steps are performed:



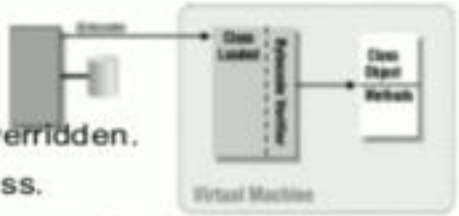
Java Class Loader:



Java Bytecode verifier:

Bytecode verifier

- No interface, users/coders cannot interact with it
- Checks:
  - The class file has the correct format
  - Final classes are not subclassed, and final methods are not overridden.
  - Every class (except for java.lang.Object) has a single superclass.
  - There is no illegal data conversion of primitive data types (e.g., int to Object)
  - There are no operand stack overflows or underflows
- Delayed bytecode verification
- Runtime verification:array bounds checking, object casting



Some Valid Class Names in Java

Class Name	Description
`Employee`	Begins with an uppercase letter
`UnderGradStudent`	Begins with an uppercase letter, contains no spaces, and emphasizes each new word with an initial uppercase letter
`InventoryItem`	Begins with an uppercase letter, contains no spaces, and emphasizes the second word with an initial uppercase letter
`Budget2014`	Begins with an uppercase letter and contains no spaces

Legal but unconventional and non-recommended class names in Java

Class Name	Description
`Undergradstudent`	New words are not indicated with initial uppercase letters, making this identifier difficult to read
`Inventory_Item`	Underscore is not commonly used to indicate new words
`BUDGET2014`	Using all uppercase letters for class identifiers is not conventional
`budget2014`	Conventionally, class names do not begin with a lowercase letter

Some illegal class names in Java

Class Name	Description
`Inventory Item`	Space character is illegal in an identifier
`class`	`class` is a reserved word
`2014Budget`	Class names cannot begin with a digit
`phone#`	The number symbol (#) is illegal in an identifier

Java Reserved Keywords

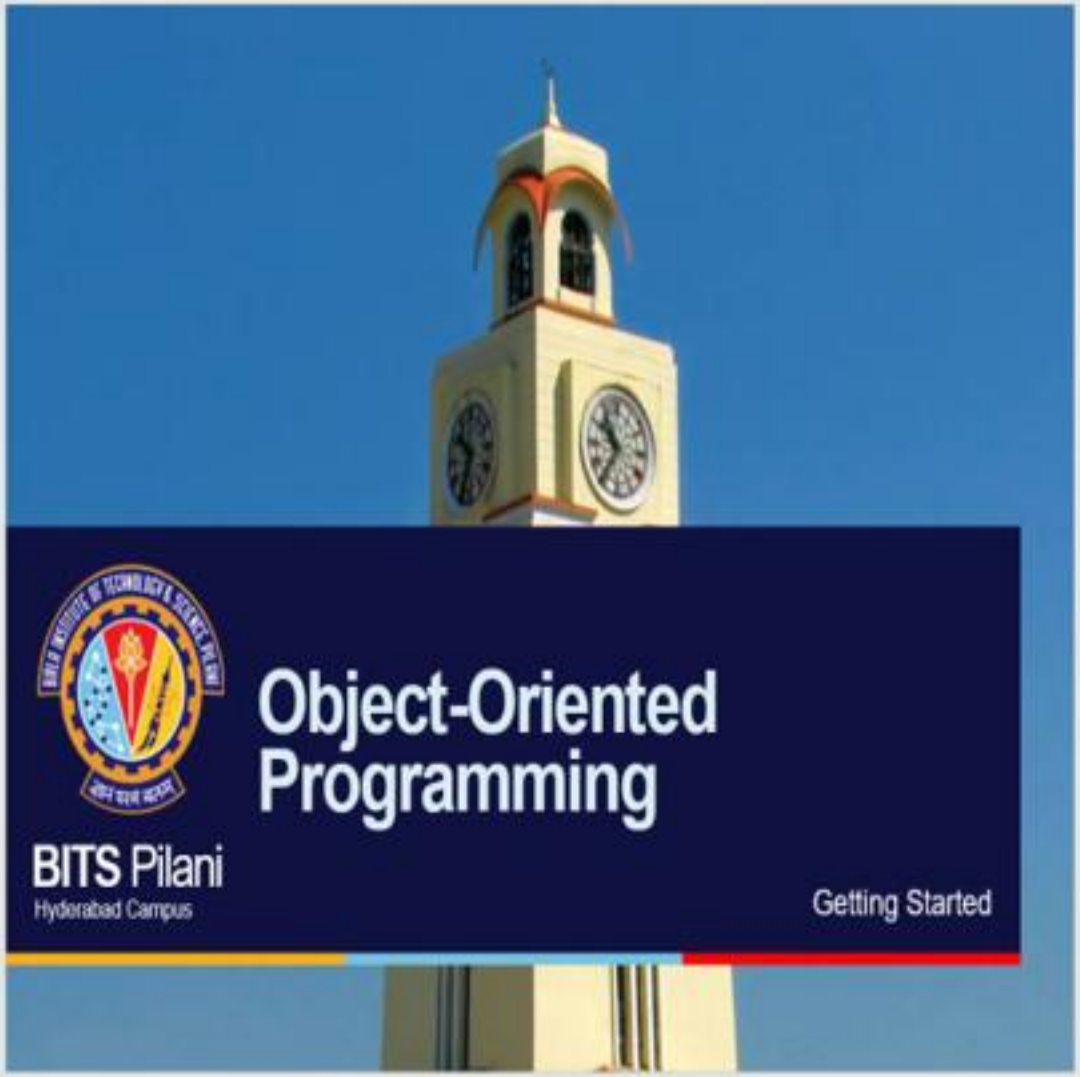
abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

Adding Comments to a Java Class

- There are three types of comments in Java:
  - ***Line comments*** start with two forward slashes (//) and continue to the end of the current line. A line comment can appear on a line by itself or at the end (and to the right) of a line following executable code. Line comments do not require an ending symbol.
  - ***Block comments*** start with a forward slash and an asterisk (/*) and end with an asterisk and a forward slash (*/). A block comment can appear on a line by itself, on a line before executable code, or on a line after executable code. Block comments also can extend across as many lines as needed.
  - Javadoc comments are a special case of block comments called documentation comments because they are used to automatically generate nicely formatted program documentation with a program named javadoc. Javadoc comments begin with a forward slash and two asterisks (/**) and end with an asterisk and a forward slash (*/).

Acknowledgement

- I deeply acknowledge and appreciate the contributions of the authors of Java Programming by Joyce Farrell and Java Complete Reference.

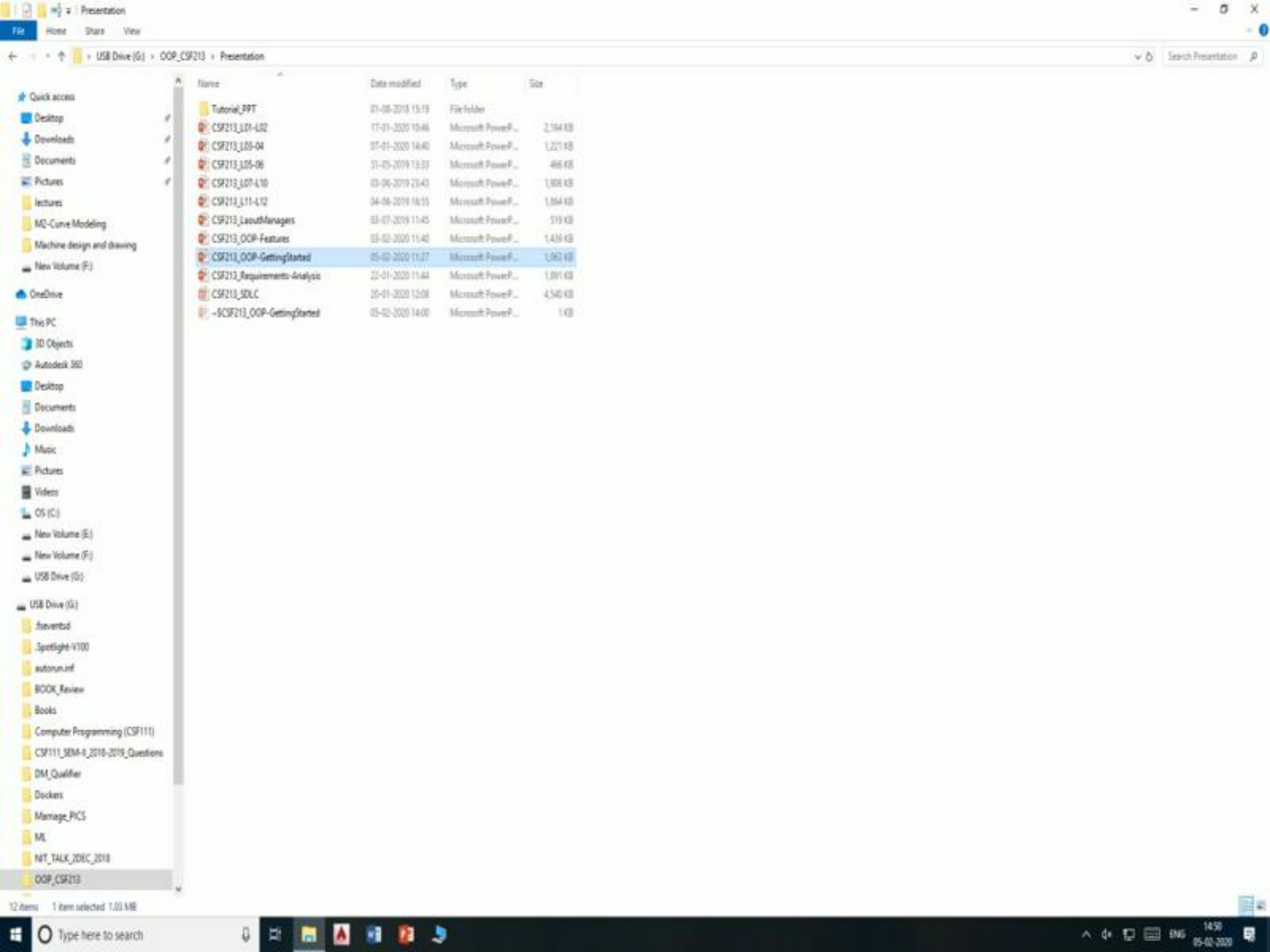


Click to add notes

Acknowledgement

- I deeply acknowledge and appreciate the contributions of the authors of Java Programming by Joyce Farrell and Java Complete Reference.

Click to add notes


Presentation
File
Home
Share
View
USB Drive (G:) › OOP_CSF213 › Presentation
Search Presentation
Quick access
Desktop
Downloads
Documents
Pictures
lectures
MC-Curve Modeling
Machine design and drawing
New Volume (F:)
OneDrive
This PC
3D Objects
Autodesk 360
Desktop
Documents
Downloads
Music
Pictures
Videos
OS (C:)
New Volume (E:)
New Volume (F:)
USB Drive (G:)
USB Drive (G:)
.fseventsd
.Spotlight-V100
autorun.inf
BOOK_Review
Books
Computer Programming (CSF111)
CSF111_SEM-II_2018-2019_Questions
DM_Qualifier
Dockers
Marriage_PICS
ML
NIT_TALK_2DEC_2018
OOP_CSF213
Name
Date modified
Type
Size
Tutorial_PPT
01-08-2018 15:19
File folder
CSF213_L01-L02
17-01-2020 10:46
Microsoft PowerP...
2,164 KB
CSF213_L03-04
07-01-2020 14:40
Microsoft PowerP...
1,221 KB
CSF213_L05-06
31-05-2019 15:33
Microsoft PowerP...
486 KB
CSF213_L07-L10
03-06-2019 23:43
Microsoft PowerP...
1,906 KB
CSF213_L11-L12
04-06-2019 16:55
Microsoft PowerP...
1,864 KB
CSF213_LaoutManagers
03-07-2019 11:45
Microsoft PowerP...
519 KB
CSF213_OOP-Features
03-02-2020 11:40
Microsoft PowerP...
1,439 KB
CSF213_OOP-GettingStarted
05-02-2020 11:27
Microsoft PowerP...
1,062 KB
CSF213_Requirements-Analysis
22-01-2020 11:44
Microsoft PowerP...
1,891 KB
CSF213_SDLC
20-01-2020 12:08
Microsoft PowerP...
4,540 KB
~$CSF213_OOP-GettingStarted
05-02-2020 14:00
Microsoft PowerP...
1 KB
12 items
1 item selected 1.03 MB
Type here to search
ENG
14:50
05-02-2020


Ink-Squared Deluxe
dump
Recycle Bin
Type here to search
ENG
14:50
05-02-2020


dump
Recycle Bin
Type here to search
ENG
14:52
05-02-2020