Basic Overview of LAB

Getting Started with Java Programming Language

- A Simple Java Application
- Compiling Programs
- Executing Applications

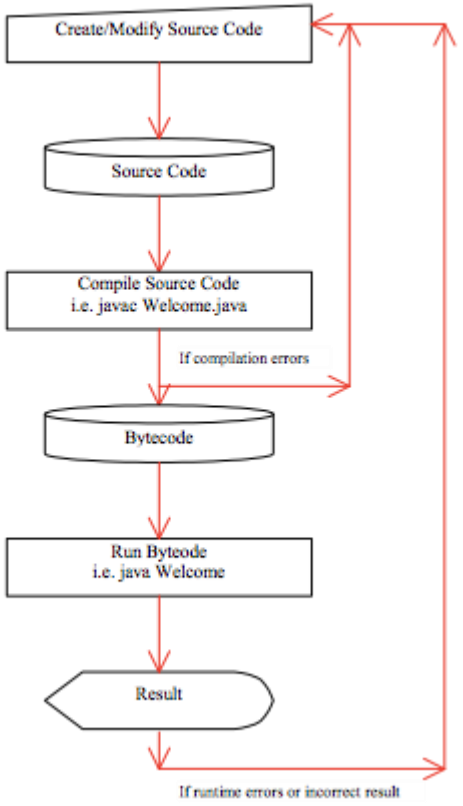
A Simple Java Application

```
/*
This is a simple Java program.
Call this file "Example.java".
NOTE: Class name and file name must be same
 */
class Example {
 // Your program begins with a call to main().
public static void main(String args[]) {
System.out.println("This is a simple Java program.");
}
}
```

Compiling Application

On Command Line:

C:/>javac Example.java



Executing Application

On Command Line:

C:/>java Example

A Closer Look at Hello World Application

/*

This is a simple Java program.

Call this file "Example.java".

NOTE: Class name and file name must be same

*/

This is comment, which describes about the programs.

Ignore by compiler

3 types of comments

Above shown is multiline comment(starts with /* and ends with */)

class Example {

- class is a keyword to declare that a new class is being defined
- Example is a Identifier that is name of class.
- Entire class definition is enclosed between { and }.

// Your program begins with a call to main().

- Above statement is single line comment.
- Single line comment starts with // and ends at the end of the line

public static void main(String args[]) {

- This line begins the main method.
- All Java applications begin execution by calling main( ).
- The full description of this statement will involve a whole chapter,so we will discussed about it later chapter.

System.out.println("This is a simple Java program.");

- This line outputs the string “This is a simple Java program.” followed by a new line on the screen.
- All statement ends with ; in java.

Example

```
class Example {


public static void main(String args[]) {

int i = 0;

}

}
```

Q. How many Identifier are there?