LAB ACTIVITY 1:

Introduction to Java Programming

Learning Outcomes:

By the end of this laboratory session, you should be able to:

1. Identify the features of Java programming language.
2. Apply programming style and documentation in Java.
3. Write, compile and run the Java programs.
4. Troubleshoot the source code to identify the errors in Java.

Hardware/Software: Computer with JDK latest version.

# Activity 1A



Activity Outcome : Identify the anatomy of the Java Program.

Procedure:

1. Analyze the HelloWorldApp program
2. Complete the following table with the correct line number and code of Java program:
3. /\* Activity 1A \*/
4. class HelloWorldApp {

3.

4.

5.

6.

7. }

public static void main(String[] args)

{

System.out.println("Hello World!");

}

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Components** | **Line No** | **Corresponding Code** |
| 1. | Comments | 1. | /\* Activity 1A \*/ |
| 2. | Class name | 2. | class HelloWorldApp |
| 3. | Access modifier | 3. | public |
| 4. | Argument | 5. | ("Hello World!"); |
| 5. | Static method | 3. | public static void main(String[] args) |

Activity Outcome: Identify programming style and documentation in Java. Write java program with programming style and documentation.

**Activity 1B**



Procedure:

**Step 1:** Open Notepad and type the following code:

/\* Activity 1B \*/

class Act1B // define class name

{ // open curly brace the beginning of class block

// main program where the program start public static void main(String[] args)

{ // open curly brace for main block

//statement to print string

System.out.println("Hello, my name is \_<YOUR NAME>."); System.out.println("\tI’m from .");

} //close curly brace for main block

} //end of class with close curly brace for body of block class

**Step 2:** Save, compile, and run the program. Save the program as Act1B.java. Write the output in the area below.

**Edited Code**

Graphical user interface, text, application

Description automatically generated

**Output:**

Text

Description automatically generated

# Activity 1C



Activity Outcome: Identify programming style and documentation in Java program. Write java program with programming style and do documentation.

Procedures :

**Step 1:** Open Notepad and type the following code:

/\* Activity 1C \*/

public class Act3F { public static void main(String[] args) {

int number1 = 11; int number2 = 22; int number3 = 33; int number4 = 44; int number5 = 55; int sum;

sum = number1 + number2 + number3 + number4 + number5; System.out.print("The sum is "); System.out.println(sum); }}

**Step 2:** Add the appropriate comments and comment style, proper indention and spacing and block styles

**Edited Code**

**Text

Description automatically generated**

**Step 3:** Save, compile, and run the program. Save the program as Act1C.java. Write the output in the area below.

## Output:



Activity Outcome: Identify programming errors in Java program. Identify the errors and modify the program to fix the error(s).

**Activity 1D**

/\* Activity 1D \*/

Class Act1D

{

public static void Main(String[] args) [

System.out.println ("I Love Java Programming!”); System.out.println ("I know how to write Java Program!); System.out.print (“It is simple and easy.”)

)

}

**Edited Code**

**Graphical user interface, text, application

Description automatically generated**

**Output**

**Text

Description automatically generated**

# Activity 1E (CLO2)



Activity Outcome: Identify programming errors in Java program. Identify the errors and modify the program to fix the error(s).

/\* Activity 1E //

class Act1E

{

public static void Main(String[] args)

{

double price1=50.65; double total = price1/0;

System.out.println (“Total price is “+total);

}

}

Write the correct program below:

**Program:**

Graphical user interface, text, application

Description automatically generated

**Output**

****

Activity Outcome: Identify programming errors in Java.

**Activity 1F (CLO2)**

Identify syntax errors in Java, implements programming style and documentation.

Procedures :

**Step 1 :** Open Notepad and type the following code:

/\* Activity 1F \*/

Class Act3I {

public Static void main(String[] args) { Sytem.out.prntln("Hello World!");

}

}

**Step 2 :** Save the program as Act1F.java in the working directory, compile and run the program.

**Step 3 :** Observe the output.

**Text

Description automatically generated**

**Step 4:** Identify the syntax error and correct the syntax errors.

**Edited Code:**

Text

Description automatically generated

**Step 5:** Save your program, compile and run the program again.

**Output:**

****

# 

# Activity 1G (CLO2)



Activity Outcome: Identify programming errors in Java.

Identify logic errors in Java, implements programming style and documentation. Procedures :

**Step 1 :** Open Notepad and type the following code:

/\* Activity 1G \*/

class Act1G {

public static void main(String[] args) { int mark = 59;

if (mark < 50) // programmer mistake but syntaxly correct System.out.println(“PASSED”);

else

System.out.println(“FAILED”);

}

}

**Step 2 :** Save the program as Act1G.java in the working directory, compile and run the program.

**Step 3 :** Observe the output.

**Text

Description automatically generated**

**Step 4:** Identify the logic error and correct the logic errors.

**Edited Code:**

**Text

Description automatically generated**

**Step 5 :** Save your program, compile and run the program again.

## Output:

## 