LAB ACTIVITY 8:

ABSTRACT CLASS AND INTERFACE

Learning Outcomes

This Lab sheet encompasses 6 activities (Activity 8A, 8B, 8C, 8D,8E and 8F).

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

* Implement abstract classes in Java program
* Classify the built-in interface class in Java Program
* Create Java program using following classes:
  + Implementing interface

**Activity 8A**

Briefly explain abstract class in Java programs.

Abstract class is a restricted class that cannot be used to create an object. If we want to access abstract class’s code, it must be inherited from another class

Abstract Class

State the rules to create abstract class and method

Rules of Abstract Class and Method

* Must be declared with an abstract keyword
* Cannot be instantiated
* It can have final method
* It can have abstract and non - abstract method
* It can have constructors and static method

**Activity 8B**

Activity Outcome: Illustrates the use of Abstract classes in java

Procedure:

**Step 1**: Type the following code

**Step 2:** Save the program

**Step 3:** Compile and execute the program

**Step 4:** Write the output

|  |
| --- |
| abstract class Arithmetic  {  int a,b;  abstract void calc();  }  class Add extends Arithmetic  {  void calc()  {  System.out.println(“The sum is “ + (a+b));  }  }  class Sub extends Arithmetic  {  void calc()  {  System.out.println(“The difference is “ + (a-b));  }  }  class Abst  {  public static void main (String args[])  {  Add obj1 = new Add();  obj1.a = 10;  obj1.b = 6;  obj1.calc();  Sub obj2 = new Sub();  obj2.a = 10;  obj2.b = 6;  obj2.calc();  }  } |

**Code**

Text

Description automatically generated

**Output**



**Activity 8C**

Activity Outcome: Illustrates the use of Abstract classes in java

Procedure :

**Step 1:** Key-in below program.

class Bike12{

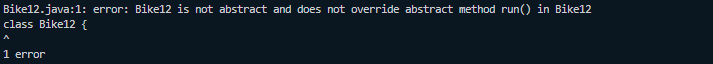
abstract void run();

}

Bike12.java

**Step 2 :** Observe the output.

Output:



**Step 3:** How to correct the above error?

Correct error above.

Code

Text

Description automatically generated

ACTIVITY 8D

The following example illustrates the use of Abstract classes in java

Procedure :

**Step 1**: Type the following code

**Step 2:** Compile the program

**Step 3:** Identify the missing syntax and correct them

**Step 4:** Save the program

**Step 5:** Compile and execute the program

**Step 6:** Write the output

abstract class Shape // abstract class

{

abstract void draw(); // abstract method

}

class Triangle extends Shape //sub class Rectangle

{

void draw(){System.out.println("drawing triangle");}

// Abstract method implemented in a subclass

}

class Circle1 extends Shape

{

}

class TestAbstraction1

{

public static void main(String args[])

{

Shape s=new Triangle();

s.draw();

}

}

Code

Text

Description automatically generated

**Output:**



**Activity 8E**

Activity Outcome: Demonstrates the implement of polymorphism and interface in Java Program.

**Procedure:**

**Step 1:** Type the following program.

|  |
| --- |
|  |

**Step 2:** Compile and execute the program

A screenshot of a computer

Description automatically generated with medium confidence

**Step 3:** Write the output

A screenshot of a computer

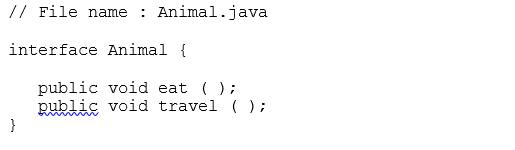
Description automatically generated with medium confidence

**Activity 8F**

Activity Outcome Create Java program using implementing interface

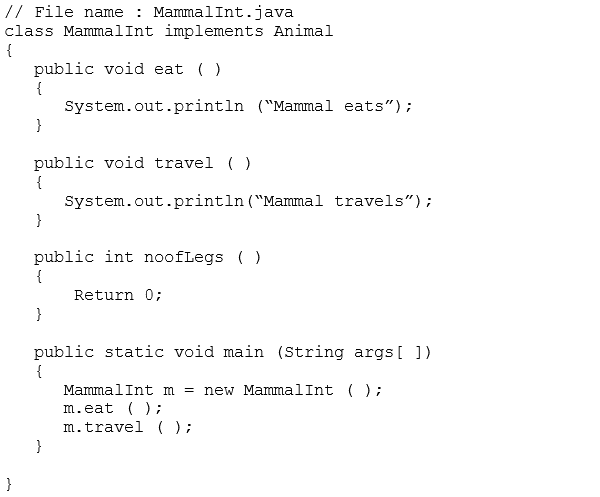
**Procedure:**

**Step 1:** Type the following program.



**Step 2:** Save the program as Animal.java

**Step 3:** Open new notepad. Type the following program.



**Step 2:** Compile and execute the program

A screenshot of a computer

Description automatically generated with medium confidence

**Step 3:** Write the output

