|  |
| --- |
| Experiment No. 11 |
| Program on Abstract class and abstract methods. |
| Date of Performance: |
| Date of Submission: |

**Aim:-** To Program on abstract class and abstract methods.

**Objective** **:-** To implement the concept of abstract class and abstract methods through java program for calculating area the geometrical figure.

**Theory :-** An exception (or exceptional event) is a problem that arises during the execution of a program. When an Exception occurs the normal flow of the program is disrupted and the program/Application terminates abnormally, which is not recommended, therefore, these exceptions are to be handled.

An exception can occur for many different reasons. Following are some scenarios where an exception occurs.

* A user has entered an invalid data.
* A file that needs to be opened cannot be found.
* A network connection has been lost in the middle of communications or the JVM has run out of memory.

The code which is prone to exceptions is placed in the try block. When an exception occurs, that exception occurred is handled by catch block associated with it. Every try block should be immediately followed either by a catch block or finally block.

**Code:-**

1. **abstract** **class** Shape{
2. **abstract** **void** draw();
3. }
4. //In real scenario, implementation is provided by others i.e. unknown by end user
5. **class** Rectangle **extends** Shape{
6. **void** draw(){System.out.println("drawing rectangle");}
7. }
8. **class** Circle1 **extends** Shape{
9. **void** draw(){System.out.println("drawing circle");}
10. }
11. //In real scenario, method is called by programmer or user
12. **class** TestAbstraction1{
13. **public** **static** **void** main(String args[]){
14. Shape s=**new** Circle1();//In a real scenario, object is provided through method, e.g., getShape() method
15. s.draw();
16. }
17. }

**Code:**

abstract class Language {

public void display() {

System.out.println("This is Java Programming");

}

}

class Main extends Language {

public static void main(String[] args) {

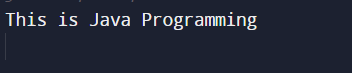
Main obj = new Main();

obj.display();

}

}

**Output :**

****

**Conclusion:**

abstract classes and methods are fundamental to object-oriented programming, allowing for the creation of structured class hierarchies and method contracts. Abstract classes provide a way to define a common structure with a mix of concrete and abstract methods, while abstract methods ensure that specific behaviors are implemented in derived classes.