Shreyas Sawant
D7A - 55

AIM: To implement abstract class in Java

THEORY:

- Abstract Classes

Abstraction is a process of hiding the implementation details and showing only functionality to the user. Abstraction lets you focus on what the object does instead of how it does it.

A class which is declared as abstract is known as an abstract class. It can have abstract and non-abstract methods. It needs to be extended and its method implemented. It cannot be instantiated.. It can ~~k~~also have constructors and static methods and also it can have final methods which will force the subclass not to change the body of the method.

For eg:

```
abstract class Bike {
  Bike ()                     // Constructor
 { System.out.printfn ("Bike is started"); }
  abstract void run ();        // abstract method, i.e. has no body
  void change Gear ()          // non-abstract method
 { System.out.println ("Gear changed"); }

class Bull extends Bike {
  void run(){ System.out.println ("Running safely"); }
}

class Demo {
public static void main (String args[]) {
  ~~Bik~~ Bike ob= new Bull ();
      ob.run ();
      ob.changeGear ();
} }
```

Output:

```
Bike is started
Running safely
Gear changed.
```

- **Importemce of abstract keyword**

'abstract' is a non-access modifier in java applicable for classes, methods but not variables. The 'abstract' keyword is used to achie abstraction in Java.

Classes declared as abstract are used to contain abstract methods however they can also contain non-abstract methods. The methods which are declared with abstract keyword and doesn't have any implementation is known as an abstract method. The rules of 'abstract' keyword are:

i) An. abstract keyword cannot be used with variables and constructors.

ii) If a class is abstract, it cannot be instantiated.

iii) We cannot use abstract keyword with final and cannot declare abstract methods as private or static.

iv) If a class extends the abstract class, it must also implement at least one of the abstract method

v) The abstract class can also be used to provide some implementation of the interface. In such cases, the end user may not be forced to override all the methods of interface.

- Difference between abstract Class and interface

Abstract Class	Interface
1) Abstract class have abstract and non-abstract methods.	1) Interface can only have abstract methods. It can have default and static methods also.
2) Abstract class ~~ha~~ doesn't support multiple inheritance.	2) Interface supports multiple inheritanc
3) Absbract class can have final, non-final static and non-static variables.	3) Interface has only static and final variables
4) Abstract class can provide the implementation of interface.	4) Interface can't provide the implementatio of abstract class.
5) The abstract keyword is used to declare abstract class.	5) The interface keyword is used to declare interface.
6) An abstract class can extend another class and implement multiple interfaces.	6) An interface can be implemented using keyword implements.

7) Example:
public abstract class Shape {
abstract void draw ();
}
	7) Example
	public interface Drawable {
	void draw();
	}

CONCLUSION:

Errors encountered :

1) Declared variable dim_1, dim_2, r as abstract :

```
abstract double dim_1, dim_2, r;
```

Solution Not using abstract keyword while declaring variables solves the error.

2) ~~Ab~~Incorrect syntax of abstract method area ():

```
abstract double area ()
```

Solution While declaring abstract methods ';' is necessarry after method name

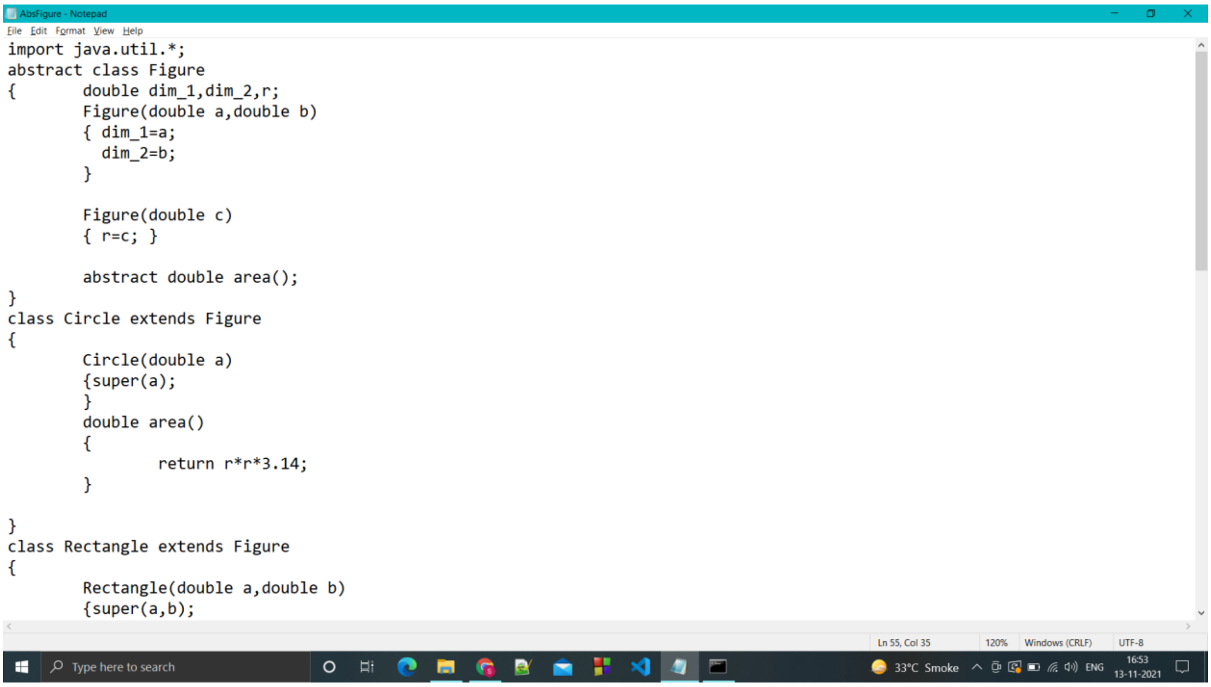
```
abstract double area ();
```

LAB 6: ABSTRACT CLASS IN JAVA

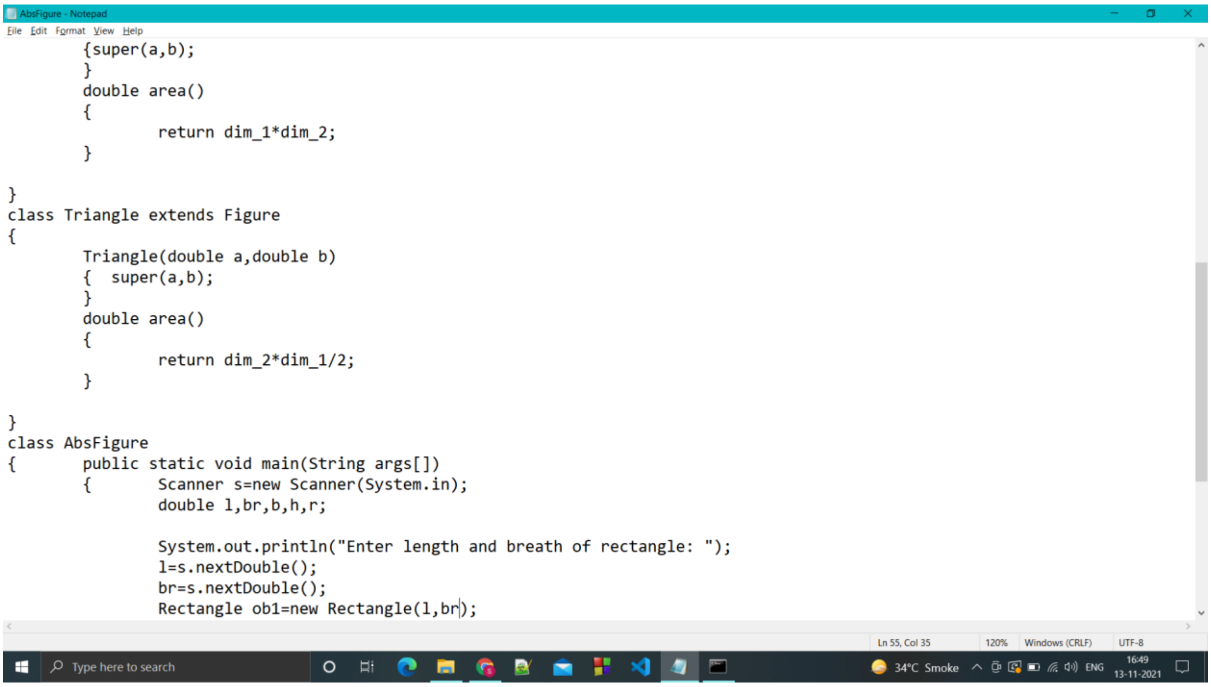
Name: Shreyas Sawant Div: D7A Roll No.: 55

Q .1 Take a class called Figure consisting of abstract method area(). Inherit this figure class through various other classes like Circle, Rectangle and Triangle which will redefine the function called area() as per the required formula.

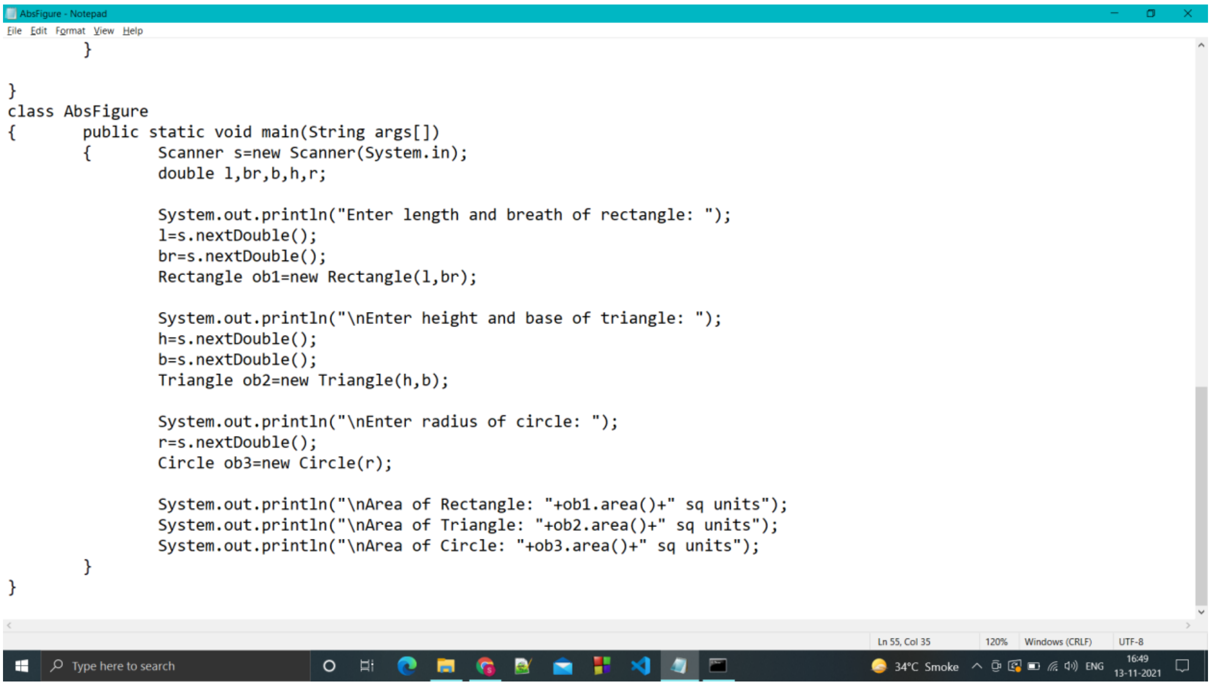
CODE:



```
import java.util.*;
abstract class Figure
{       double dim_1,dim_2,r;
        Figure(double a,double b)
        { dim_1=a;
          dim_2=b;
        }

        Figure(double c)
        { r=c; }

        abstract double area();
}
class Circle extends Figure
{
        Circle(double a)
        {super(a);
        }
        double area()
        {
                return r*r*3.14;
        }

}
class Rectangle extends Figure
{
        Rectangle(double a,double b)
        {super(a,b);
```

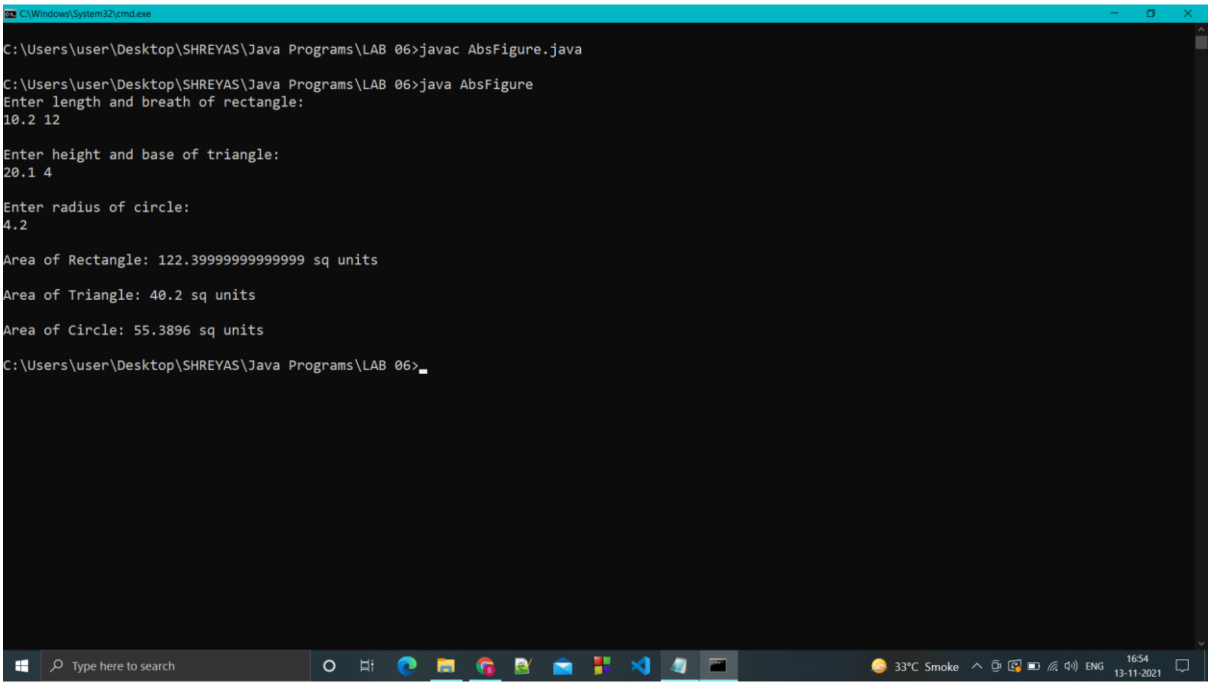


```
        {super(a,b);
        }
        double area()
        {
                return dim_1*dim_2;
        }

}
class Triangle extends Figure
{
        Triangle(double a,double b)
        {  super(a,b);
        }
        double area()
        {
                return dim_2*dim_1/2;
        }

}
class AbsFigure
{       public static void main(String args[])
        {       Scanner s=new Scanner(System.in);
                double l,br,b,h,r;

                System.out.println("Enter length and breath of rectangle: ");
                l=s.nextDouble();
                br=s.nextDouble();
                Rectangle ob1=new Rectangle(l,br);
```



```
        }

}
class AbsFigure
{       public static void main(String args[])
        {       Scanner s=new Scanner(System.in);
                double l,br,b,h,r;

                System.out.println("Enter length and breath of rectangle: ");
                l=s.nextDouble();
                br=s.nextDouble();
                Rectangle ob1=new Rectangle(l,br);

                System.out.println("\nEnter height and base of triangle: ");
                h=s.nextDouble();
                b=s.nextDouble();
                Triangle ob2=new Triangle(h,b);

                System.out.println("\nEnter radius of circle: ");
                r=s.nextDouble();
                Circle ob3=new Circle(r);

                System.out.println("\nArea of Rectangle: "+ob1.area()+" sq units");
                System.out.println("\nArea of Triangle: "+ob2.area()+" sq units");
                System.out.println("\nArea of Circle: "+ob3.area()+" sq units");
        }
}
```

OUTPUT:



```
C:\Users\user\Desktop\SHREYAS\Java Programs\LAB 06>javac AbsFigure.java

C:\Users\user\Desktop\SHREYAS\Java Programs\LAB 06>java AbsFigure
Enter length and breath of rectangle:
10.2 12

Enter height and base of triangle:
20.1 4

Enter radius of circle:
4.2

Area of Rectangle: 122.39999999999999 sq units

Area of Triangle: 40.2 sq units

Area of Circle: 55.3896 sq units

C:\Users\user\Desktop\SHREYAS\Java Programs\LAB 06>_
```