Name: Prathamesh Baban Harad

Inheritance and method overriding:

Inheritance and method overriding:

1. Edit this code so the class Beagle is a subclass of the Dog class. When you run the code it should print “woof!” and then “arf arf”

```
public class Dog
{
public void speak()
{
System.out.println("woof!");
}
public static void main(String[] args)
{
Dog b = new Beagle();
b.speak();
}
}
class Beagle extends Dog
{
public void speak()
{
super. speak();
System.out.println("arf arf");
}
}
```

2. Add an equals method to this class that returns true if the current Dog and passed Dog have the same name. The code should print false twice then true twice.

```
public class Dog
{
private String name;
public Dog(String name)
{
this.name = name;
}
public boolean equals(Object other)
{
//
}
public static void main(String[] args)
{
Dog d1 = new Dog("Rufus");
Dog d2 = new Dog("Sally");
Dog d3 = new Dog("Rufus");
Dog d4 = d3;
System.out.println(d1.equals(d2));
System.out.println(d2.equals(d3));
System.out.println(d1.equals(d3));
System.out.println(d3.equals(d4));
}
}
```

Solution::

```
class Dog
{
private String name;
public Dog(String name)
{
this.name = name;
}
@Override
public boolean equals(Object obj)
{
Dog d12 = (Dog)obj;
return this.name.equals(d12.name);
}
}
public class MyClass{
public static void main(String[] args)
{
Dog d1 = new Dog("Rufus");
Dog d2 = new Dog("Sally");
Dog d3 = new Dog("Rufus");
Dog d4 = d3;
System.out.println(d1.equals(d2));
System.out.println(d2.equals(d3));
System.out.println(d1.equals(d3));
System.out.println(d3.equals(d4));
}
}
```

Abstract class:

A Java abstract class is a class that can't be instantiated. That means you cannot create new instances of an abstract class. It works as a base for subclasses.Following is an example of abstract class:

```
abstract class Book{
String title;
abstract void setTitle(String s);
String getTitle(){
return title;
}
}
```

If you try to create an instance of this class like the following line you will get an error:

Book new_novel=new Book();
You have to create another class that extends the abstract class. Then you can create an instance of the new class.
Notice that setTitle method is abstract too and has no body. That means you must implement the body of that method in the child class.
In the editor, we have provided the abstract Book class and a Main class. In the Main class, we created an instance of a class called MyBook. Your task is to write just the MyBook class.
Your class mustn't be public.
Sample Input
A tale of two cities
Sample Output
The title is: A tale of two cities
Solution::

```
import java.util.*;
abstract class Book{
String title;
abstract void setTitle(String s);
String getTitle(){
return title;
}
}
class MyBook extends Book {
@Override
void setTitle(String s) {
title = s;
}
}
public class Main{
public static void main(String []args){
Scanner sc=new Scanner(System.in);
String title=sc.nextLine();
MyBook new_novel=new MyBook();
new_novel.setTitle(title);
System.out.println("The title is: "+new_novel.getTitle());
sc.close();
}}
```