TEST 1

1. Create one superclass HillStations and three subclasses Manali, Mussoorie, Gulmarg. Subclasses extend the superclass and override its location() and famousFor() method. i.call the location() and famousFor() method by the Parent class', i.e. Hillstations class. As it refers to the base class object and the base class method overrides the superclass method; the base class method is invoked at runtime. ii.call the location() and famousFor() method by the all subclass',and print accordingly.

SOLUTION:

```
class HillStations
{
    void location()
     {
       System.out.println("Located in the mountains.");
     }

    void famousFor()
     {
       System.out.println("Famous for natural beauty and scenic views.");
     }
}

class Manali extends HillStations {
    @Override
    void location()
     {
       System.out.println("Located in Himachal Pradesh, India.");
     }

    @Override
    void famousFor()
     {
       System.out.println("Famous for snow-capped mountains, adventure sports,
and Hadimba Temple.");
     }
}
```

```
class Mussoorie extends HillStations {
    @Override
    void location()
     {
        System.out.println("Located in Uttarakhand, India.");
     }

    @Override
    void famousFor()
     {
        System.out.println("Famous for scenic views, Kempty Falls, and Mall Road.");
     }
}

class Gulmarg extends HillStations {
    @Override
    void location()
     {
        System.out.println("Located in Jammu and Kashmir, India.");
     }

    @Override
    void famousFor()
     {
        System.out.println("Famous for skiing, Gondola ride, and lush meadows.");
     }
}

public class Main {
    public static void main(String[] args) {
        HillStations hillStation = new HillStations();
        hillStation.location();
        hillStation.famousFor();
        Manali manali = new Manali();
        Mussoorie mussoorie = new Mussoorie();
        Gulmarg gulmarg = new Gulmarg();
        manali.location();
        manali.famousFor();
        mussoorie.location();
        mussoorie.famousFor();
```

```
        gulmarg.location();
        gulmarg.famousFor();
    }
}
```

OUTPUT:

```
Located in the mountains.
Famous for natural beauty and scenic views.
Located in Himachal Pradesh, India.
Famous for snow-capped mountains, adventure sports, and Hadimba Temple.
Located in Uttarakhand...
```

2. Write a Java program that demonstrates method overriding by creating a superclass called Animal and two subclasses called Dog and Cat. ● The Animal class should have a method called makeSound(), which simply prints "The animal makes a sound." ● The Dog and Cat classes should override this method to print "TheCat/The dog meows/barks" respectively. ● The program should allow the user to create and display objects of each class.

SOLUTION:

```
public class AnimalDemo {

    public static void main(String[] args) {
        Animal animal = new Animal();
        Dog dog = new Dog();
        Cat cat = new Cat();
        System.out.println("Animal sound: " + animal.makeSound());
        System.out.println("Dog sound: " + dog.makeSound());
        System.out.println("Cat sound: " + cat.makeSound());
    }
}

class Animal {
    public String makeSound() {
        return "The animal makes a sound.";
    }
}

class Dog extends Animal {
    @Override
```

```
    public String makeSound() {
        return "The dog barks.";
    }
}

class Cat extends Animal {
    @Override
    public String makeSound() {
        return "The cat meows.";
    }
}
```

OUTPUT:

```
Animal sound: The animal makes a sound.
Dog sound: The dog barks.
Cat sound: The cat meows.
```

3. Write code to determine if the string is a palindrome.
   input string: Madam.

   SOLUTION:

```
public class PalindromeChecker {

    public static boolean isPalindrome(String text) {
        String processedText = text.replaceAll("[^a-zA-Z0-9]", "").toLowerCase();
        int left = 0;
        int right = processedText.length() - 1;
        while (left < right) {
            if (processedText.charAt(left) != processedText.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true;
    }

    public static void main(String[] args) {
        String input = "Madam";
        if (isPalindrome(input)) {
```

```
            System.out.println(input + " is a palindrome.");
        } else {
            System.out.println(input + " is not a palindrome.");
        }
}
}
```

OUTPUT:

Madam is a palindrome.

4. You need to find and print all the unique characters in a given string.
   input string: java.

SOLUTION:

```
import java.util.HashSet;

public class UniqueCharactersFinder {
    public static void main(String[] args) {
        String input = "java";
        String uniqueChars = findUniqueCharacters(input);
        System.out.println("Unique characters in the input string: " + uniqueChars);
    }

    public static String findUniqueCharacters(String str) {
        HashSet<Character> uniqueSet = new HashSet<>();
        StringBuilder result = new StringBuilder();

        for (char ch : str.toCharArray()) {
            if (!uniqueSet.contains(ch)) {
                uniqueSet.add(ch);
                result.append(ch);
            }
        }

        return result.toString();
    }
}
```

OUTPUT:

Unique characters in the input string: jv