1. Method Overloading: Write a class Calculator with overloaded methods add(). Implement add() methods that take:
   - Two integers
   - Two double values
   - Three integers
   - A variable number of integers

Code:-

```
package MyPackage;
//creating a class calculator
Public class Calculator
{
  //add method takes two integer
  Public int add(int a, int b) {
    Return a + b;
  }

  //add method takes two double
  Public double add(double a, double b) {
    Return a + b;
  }

  //add method takes three integer
  Public int add(int a, int b, int c) {
    Return a + b + c;
  }

  // Method to add a variable number of integers
  Public int add(int … numbers) {
    Int sum=0;
    For (int number : numbers) {
      Sum += number;
    }
    Return sum;
  }

  Public static void main(String[] args)
  {
    //creating the object of calculator class
    Calculator c = new Calculator();

    //Testing the add methods
    System.out.println(“Add two integer value (2 + 3) : “ + c.add(2, 3));
    System.out.println(“Add two double value (2.5 + 3.5) : “ + c.add(2.5,
3.5));
    System.out.println(“Add three integer value (2 + 3 + 4) : “ + c.add(2,
3, 4));
    System.out.println(“Add variable number of integers (1, 2, 3, 4, 5): “
+ c.add(1, 2, 3, 4, 5));
  }
}
```

Output:-

```
Add two integer value (2 + 3) : 5
Add two double value (2.5 + 3.5) : 6.0
Add three integer value (2 + 3 + 4) : 9
Add variable number of integers (1, 2, 3,
    4, 5): 15
```

2.Super Keyword: Create a class Person with a constructor that accepts and sets name and age.

- Create a subclass Student that adds a grade property and initializes name and age using the super keyword in its constructor.
- Demonstrate the creation of Student objects and the usage of super to call the parent class constructor.

Code:-

```
package MyPackage;
//creating a super class Parent
Class Person
{
  String name;
  Int age;

  //super class constructor
  Public Person(String name, int age) {
    This.name=name;
    This.age=age;
    System.out.println(“Person class constructor called.”);
  }

  //super class method
  Public void getDetails() {
    System.out.println(“Name : “ + name);
    System.out.println(“Age : “ + age);
  }
}

//subclass Student extend super class Parent
Class Student extends Person
{
  Char grade;

   //subclass constructor
  Public Student(String name, int age, char grade) {
    Super(name, age);
    This.grade=grade;
    System.out.println(“Student class constructor called.”);
  }

  @Override //subclass override parent class method
```

```
  Public void getDetails() {
    System.out.println(“Name : “ + name);
    System.out.println(“Age : “ + age);
    System.out.println(“Grade : “ + grade);
  }
}

Public class SuperKeywordDemo
{
  Public static void main (String[] args)
  {
    //creating object of subclass Student
    Student obj=new Student(“Pawan”, 21, ‘B’);

    //calling subclass method
    Obj.getDetails();
  }
}
```

Output:-

```
Person class constructor called.
Student class constructor called.
Name : Pawan
Age : 21
Grade : B
```

3.Super Keyword: Create a base class Shape with a method draw() that prints “Drawing Shape”.

- Create a subclass Circle that overrides draw() to print “Drawing Circle”.
- Inside the draw() method of Circle, call the draw() method of the Shape class using super.draw().
- Write a main method to demonstrate calling draw() on a Circle object.

Code:-

```
package MyPackage;
//creating a super class Shape
Class Shape
{
  //super class method
  Public void draw() {
    System.out.println(“Drawing Shape”);
  }
}

//subclass Circle extend super class Shape
Class Circle extends Shape
{
  @Override //subclass method override super class method
```

```
  Public void draw() {
    Super.draw(); //calling super class method
    System.out.println("Drawing Circle");
  }
}

Public class SuperKeywordDemo
{
  Public static void main(String[] args)
  {
    //creating object of subclass Circle
    Circle c=new Circle();

    c.draw(); //calling subclass method
  }
}
```

Output:-

```
Drawing Shape
Drawing Circle
```

4.Write a Java Program to count the number of words in a String without using the Predefined method?

Code:-

```
package MyPackage;
public class CountWords
{
  Public static void main(String[] args)
  {
    //string to count words
    String str = "Hello welcome to the world of Java";

    //split method is applied on string to split string into multiple
string and length method provide the number of strings
    Int wordCount =  str.split(\\s).length;

    //print the number of words in a string
    System.out.println("Number of words in a string is: " + wordCount);
  }
}
```

Output:-

```
Number of words in a string is: 7
```

5.Write a Java Program to remove all white spaces from a String?

Code:-

```
package MyPackage;
public class WhiteSpaceRemover
{
  Public static void main(String[] args)
  {
    //string with white spaces
    String str = “H e ll o   W or l d  .”;

    //applying replaceAll method to string and replace white spaces
    String newStr = str.replaceAll(“\\s”, “”);

    //printing both the strings
    System.out.println(“String with white spaces : “ + str);
    System.out.println(“String without white spaces : “ + newStr);
  }
}
```

Output:-

```
String with white spaces : H e ll o   W or
    l d  .
String without white spaces : HelloWorld.
```

6.WAP to find occurrence of given in the given string.

Code:-

```
package MyPackage;
import java.util.Scanner;
public class SubstringChecker
{
  Public static void main(String[] args)
  {
    // Creating a Scanner object
    Scanner sc = new Scanner(System.in);

    // taking main string input from user
    System.out.print(“Enter the main string: “);
    String mainString = sc.nextLine();

    // taking substring input from user
    System.out.print(“Enter the substring to check: “);
    String substring = sc.nextLine();
```

```
    // check substring occurs in main string or not
    If (mainString.contains(substring))
    {
      System.out.println("The substring occurs in the main string.");
    } else {
      System.out.println("The substring does not occurs in the main
string.");
    }
  }
}
```

Output:-

```
Enter the main string: Hello, how are you?
Enter the substring to check: are
The substring occurs in the main string.
```

7.Write a java class to implement any 10 string methods:

● replace ● contains ● replaceAll ● indexOf ● substring ● Equals ● lastIndexOf ● startsWith ● endsWith ● EqualsIgnoreCase ● toLowerCase ● toUpperCase ● isEmpty ● Length ● split

Code:-

```
package MyPackage;
public class StringMethods
{
  Public static void main (String[] args)
  {
    String str="Hello";

    System.out.println("Replacing characters of string with replace method
: " + str.replace("lo", "p"));

    System.out.println("Demonstrating contains method : " +
str.contains("e"));

    System.out.println("Replacing all the similar character with replaceAll
method : " + str.replaceAll("l", "r"));

    System.out.println("Finding index of a character with indexOf method :
" + str.indexOf("o"));

    System.out.println("Making substring from string with substring method
: " + str.substring(0, 4));
```

```
    System.out.println("Cheking string is equal with equal method : " +
str.equals("Hello"));

    System.out.println("Using lastIndexOf method to find index : " +
str.lastIndexOf("l"));

    System.out.println("Checking string starting character with startsWith
method : " + str.startsWith("H"));

    System.out.println("Checking string ending character with endsWith
method : " + str.endsWith("o"));

    System.out.println("Using equalsIgnoreCase method : " +
str.equalsIgnoreCase("HeLlO"));

    System.out.println("LowerCase method : " + str.toLowerCase());

    System.out.println("UpperCase method : " + str.toUpperCase());

    System.out.println("Checking string is empty : " + str.isEmpty());

    System.out.println("Printing the length of string : " + str.length());

  }
}
```

Output:-

```
Replacing characters of string with replace method :
    Help
Demonstrating contains method : true
Replacing all the similar character with replaceAll
    method : Herro
Finding index of a character with indexOf method : 4
Making substring from string with substring method :
    Hell
Cheking string is equal with equal method : true
Using lastIndexOf method to find index : 3
Checking string starting character with startsWith
    method : true
Checking string ending character with endsWith method
    : true
Using equalsIgnoreCase method : true
LowerCase method : hello
UpperCase method : HELLO
Checking string is empty : false
Printing the length of string : 5
```

8.Write a java program to implement string tokenizer.

Code:-

```
package MyPackage;
// importing StringTokenizer
Import java.util.StringTokenizer;
Public class StringTokenizerDemo
{
  Public static void main(String[] args)
  {
    //creating object of StringTokenizer
    StringTokenizer st = new StringTokenizer("Hello Welcome to the world of
Java"," ");

    System.out.println("Tokens in the given string are : ");

    //print the tokens
    While (st.hasMoreTokens()) {
      System.out.println(st.nextToken());
    }
  }
}
```

Output:-

```
Tokens in the given string are :
Hello
Welcome
to
the
world
of
Java
```