1. **Write a program in Java to perform implicit and explicit type casting**

```
Package JavaBasics;
Public class TypeCastingDemo{
     Public static void main(String[] args){
  //input TypeCasting
           Int num =5;
            Double num2 = num;
              System.out.println(num2);
   //Explicit Type Casting
             double num3 = 9.8;
                int num4 =(int)num3;
                 System.out.println(num4);
      }
}
```

2.Demonstrate types of inheritance

```
 package inheritance;
  public class A extends D{
         public void day(){
                        System.out.println("Today is Tuesday");
            }
            public static void main(String[] args)


                        //Always create object of child class

                        A.a = new A();

                        a.day();

                        a.date();
            }
}
```

```
package inheritance;
  public class B extends D{
         public void month(){
                        System.out.println("It is a month");
            }
            public static void main(String[] args)


                        //Always create object of child class

                        B.b = new B();

                        b.month();
```

```
                b.date();
        }
}


package inheritance;
 public class C extends D{
      public void year(){
                System.out.println(“It is a 2023”);
        }
        public static void main(String[] args)


                //Always create object of child class

                C.c = new C();

                c.year();

                c.date();
        }
}


Package inheritance;
Public class D{

        Public void date(){

                System.out.println(“It is 27-06-2023”);

        }
}
```

3.Write a program in Java to verify the working of access modifiers

//*Public – a public property (method or a variable) is visible everywhere in the project.*

```
 package demo2;
 public class Calculator{
        public void add(int a, int b) {

                System.out.println(a+b);

        }

        Public static void main(String[] args){
                Calculator cal = new Calculator();

                cal.add(20,5);

        }
}
```

Private – a private property is visible only in the class where it is created. Outside the class it is not accessible.

```
package demo2;
 public class Calculator{
        private void add(int a, int b) {

                System.out.println(a+b);

        }

        Public static void main(String[] args){
                Calculator cal = new Calculator();

                cal.add(20,5);

        }
}
```

Protected – a protected property is visible everywhere within the package but outside the package it is visible only to child classes.

```
package demo2;
 public class Calculator{
        protected void add(int a, int b) {

                System.out.println(a+b);

        }

        Public static void main(String[] args){
                Calculator cal = new Calculator();

                cal.add(20,5);

        }
}
```

Default – a default property is visible only within the package. It cannot be accessed outside the package.

```
package demo2;
 public class Calculator{
         void add(int a, int b) {

                System.out.println(a+b);

        }
```

```
          Public static void main(String[] args){
                    Calculator cal = new Calculator();

                    cal.add(20,5);

          }
}
```

4. Write a program to demonstrate the while loop

```
package loops;
public class whileLoopDemo{
          public static void main(String[] args){

                    int PINinBankDB = 1234;

                    Scanner scanner = new Scanner(System.in);

                    System.out.println("Please enter your ATM-Pin");

                    Int PINenteredByUser = Scanner.nextInt();
                                        While(PINenteredByUser != PINinBankDB){
                                        System.out.println("The ATM pin you entered is incorrect.
                                        Please try   again!");
                                        PINenteredByUser = scanner.nextInt();
                              }

                              System.out.println("Welcome to ABC Bank");

                    }

}
```

5. Write a program to demonstrate the do while loop

```
package loops;
public class DoWhileDemo{
          public static void main(String[] args){

                    int i=0;

                    do{
                    System.out.println("Today is Thursday");
                              i++;
                    }
                    While(i<5);
          }
}
```

6. Write a program to demonstrate the for loop

```
package loops;
public class ForLoopDemo{
        public static void main(String[] args){
                for(int i=0; i<5;i++){

                        System.out.println(“Today is Thursday”);

                        System.out.println(“Tomorrow is Friday”);

                }

        }
}
```

7.Demonstrate the Classes, Objects, and Constructors

```
package demo2;
public class Car{
        private String color;

        private String engineType;

        //Constructor
        public Car(String colorOfCar , String typeOfEngine){
        color = colorOfCar;

        engineType = typeOfEngine;

        }

                Public void printCarProperties(){

                System.out.println(“color of car = “+color);

                System.out.println(“Type of Engine = “+engineType);

        }
        Public static void main(String[] args){

        Car Mercedes = new Car(“Silver”,”Petrol”);

        Car audi = new Car(“black”,”Diesel”);

        }
}
```

8. Writing a program in Java to verify implementations of collections

```
package CollectionFramework;

import java.util.ArrayList;

public class ArrayListDemo {
    public static void main(String[] args) {

        ArrayList<String> cities = new ArrayList<>();

        cities.add("Londin");
        cities.add("Paris");
        cities.add("Pune");
        cities.add("Chennai");
        cities.add("Mumbai");

        for(String t : cities) {
            System.out.println(t);
        }
        System.out.println("Total items in the array-List="+cities.size());
        System.out.println("Item at Index 1="+cities.get(1));
        System.out.println("Index of paris = "+cities.indexOf("Paris"));
        System.out.println("Is Benguluru present in the
list?"+cities.contains("Benguluru"));




    }

}

package CollectionFramework;

import java.util.HashSet;

public class HashSetDemo {
public static void main(String[] args) {

     HashSet<String> cities = new HashSet<>();

        cities.add("Londin");
        cities.add("Paris");
        cities.add("Pune");
        cities.add("Chennai");
        cities.add("Mumbai");
        cities.add("Londin");

        for(String t : cities) {
            System.out.println(t);
        }
        System.out.println("Total items in the array-List="+cities.size());
        //System.out.println("Item at Index 1="+cities.get(1));
        //System.out.println("Index of paris = "+cities.indexOf("Paris"));
        System.out.println("Is Benguluru present in the
list?"+cities.contains("Benguluru"));

}

}


package CollectionFramework;

import java.util.HashMap;

import java.util.Map;

public class HashMapDemo {

    public static void main(String[] args) {
        HashMap<String, Integer> td = new HashMap<>();

        td.put("Ram",123456);
        td.put("John",234567);
        td.put("Sam",345678);
        td.put("James",456789);
        td.put("Ram",123456);

        for(Map.Entry m : td.entrySet()) {
            System.out.println(m.getKey() + "-"+ m.getValue());
        }



    }

}
```

9.Writing a program to perform try-catch block

```
package exceptionHandling;

import java.util.Scanner;

public class Calculator {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        try {
```

```
			System.out.println("Please enter a number:");
			int num1 = scanner.nextInt();

			System.out.println("Please enter another number:");
			int num2 = scanner.nextInt();

			System.out.println("Result of division = "+(num1/num2));
		}
		catch(Exception e) {
			System.out.println("Please enter a valid input");
		}

	}

}
```

10.Writing code for a try block with parameters. Writing code for multiple catch blocks

```
package exceptionHandling;

import java.util.InputMismatchException;
import java.util.Scanner;

public class Calculator1 {

public static void main(String[] args) {

		Scanner scanner = new Scanner(System.in);

		try {
			System.out.println("Please enter a number:");
			int num1 = scanner.nextInt();

			System.out.println("Please enter another number:");
			int num2 = scanner.nextInt();

			System.out.println("Result of division = "+(num1/num2));
		}
		catch(ArithmeticException e) {
			System.out.println("Please do not enter a zero in the denominator");
		}
		catch(InputMismatchException e) {
			System.out.println("Only integer values are allowed");
		}
		catch(Exception e) {
			System.out.println("Please enter a valid input");
		}
```

```
        }

}
```

11. Writing code for finally{} block

```
package exceptionHandling;

import java.util.InputMismatchException;
import java.util.Scanner;

public class Calculator2 {

public static void main(String[] args) {

                Scanner scanner = new Scanner(System.in);

                try {
                        System.out.println("Please enter a number:");
                        int num1 = scanner.nextInt();

                        System.out.println("Please enter another number:");
                        int num2 = scanner.nextInt();

                        System.out.println("Result of division = "+(num1/num2));
                }
                catch(ArithmeticException e) {
                        System.out.println("Please do not enter a zero in the denominator");
                }
                catch(InputMismatchException e) {
                        System.out.println("Only integer values are allowed");
                }
                catch(Exception e) {
                        System.out.println("Please enter a valid input");
                }
                finally {
                        System.out.println("Hello");
                }
        }

}
```

12. Writing code for throw and throws keyword

```
package exceptionHandling;

import java.util.Scanner;
```

```
public class throwDemo {

        public static void main(String[] args) {
                Scanner scanner = new Scanner(System.in);
                System.out.println("Please enter Today's day:");
                String day = scanner.nextLine();
                if(day.toLowerCase().equals("monday")){
                        throw new NullPointerException();
                }
                else {
                        System.out.println("Weekend is approaching");
                }

        }

}
```

```
package exceptionHandling;
import java.io.File;

import java.io.FileNotFoundException;

import java.io.PrintWriter;

public class ExternalOuput{


        public static void main(String [] args)throws FileNotFoundException{
                File file = new File("/home/pusabhargavi1mp/Documents/Ouput");
                PrintWriter printWriter=new PrintWriter(file);
                printWriter.print("Hello,Today is Friday");
                printWriter.close();
        }
}
```