14. EXCEPTIONS

1.
Program to generate Arithmetic Exception without exception handling-

```
public class Excercise1
{
	public static void main(String[] args)
	{
	   int a = 100/0;   // ArithmeticException will be thrown because a number cannot be divided
by 0
	   System.out.println("Code");
	}
}
```

Output-
Exception in thread "main" java.lang.ArithmeticException: / by zero

2.
Handling the Arithmetic exception using try-catch block-

```
public class Excercise2
{
	public static void main(String[] args)
	{
	   try
	   {
		int a = 100/0;
	    }
	    catch (ArithmeticException e)
	    {
		System.out.println("Arithmetic Exception: cannot divide by 0");
	     }
	}
}
```

Output-
Arithmetic Exception: cannot divide by 0

3.

```
public class Excercise3
{
	public static void main(String[] args)
	{
	   myMethod();
	}
	static void myMethod()
	{

	   try
	   {
```

```
                int a = 100/0;
            }
            catch (ArithmeticException e)
            {
                System.out.println("Arithmetic Exception: cannot divide by 0");
             }
        }
}
```

Output-
Arithmetic Exception: cannot divide by 0

4.
Multiple catch blocks-

```
public class Excercise4
{
        public static void main(String[] args)
        {
            try{
         try{
                System.out.println("going to divide");
                int b =39/0;
            }
            catch(ArithmeticException e)
            {
               System.out.println(e);

            }
            try
            {
               int a[]=new int[5];
               a[5]=4;
            }
            catch(ArrayIndexOutOfBoundsException e)
            {
               System.out.println(e);
            }
            System.out.println("Rest of the code");
         }
         catch(Exception e)
         {
            System.out.println("exception handeled");
         }
         System.out.println("normal flow...");
        }
}
```

Output-
going to divide

```
java.lang.ArithmeticException: / by zero
java.lang.ArrayIndexOutOfBoundsException: 5
Rest of the code
normal flow...
```

5.

```
public class Excercise5
{
        public static void main(String[] args)
        {
           try
           {
                int a = 100/0;
            }
            catch (ArithmeticException e)
            {
                System.out.println("Hello World!!");
             }
        }
}
```

Outptut-

```
Hello World!!
```

6.

Creating your own exception(User-defined Exceptions)-

```
class MyException extends Exception
{
   public MyException(String s)
   {
      super(s);
   }
}

public class Excercise6
{
   public static void main(String args[])
   {
      try
      {
         throw new MyException("Exception");
      }
      catch (MyException e)
      {
         System.out.println("Caught");
         System.out.println(e.getMessage());
      }
   }
}
```

Output-
Caught
Exception

7.
Program with finally block-

```
public class Excercise7
{
   public static void main(String args[])
   {
      try{
           int data=100/2;
           System.out.println(data);
          }
          catch(NullPointerException e)
         {
            System.out.println(e);
          }
         finally
        {
            System.out.println("finally block");
         }
         System.out.println("rest of the code...");
   }
}
```

Output-
50
finally block
rest of the code...

8.
Arithmetic Exception-

```
public class Excercise8
{
          public static void main(String[] args)
          {
             try
             {
                    int a = 100/0;
              }
              catch (ArithmeticException e)
              {
                    System.out.println("Arithmetic Exception: cannot divide by 0");
               }
          }
}
```

Output-
Arithmetic Exception: cannot divide by 0

9.
ArrayIndexOutOfBoundException -

```
public class Excercise9
{
   public static void main(String args[])
   {
     try{
           int a[]=new int[5];
           a[5]=4;
         }
         catch(ArrayIndexOutOfBoundsException e)
         {
          System.out.println(e);
         }
         System.out.println("ArrayIndexOutOfBoundsException occurs");
   }
}
```

Output-

10.ClassNotFoundException-

```
public class Excercise10
{
   public static void main(String args[])
   {
     try
     {
        Class.forName("ClassA");  //class “C” is not found in classpath
     }
     catch (ClassNotFoundException ex)
     {
        ex.printStackTrace();
     }
   }
}
```

Output-
java.lang.ClassNotFoundException: ClassA

11.
FileNotFoundException -

```
import java.io.File;
```

```
import java.io.FileNotFoundException;
import java.io.FileReader;
class Excercise11
{
  public static void main(String args[])
   {
      try {
         File file = new File("E:// file.txt");
         FileReader fr = new FileReader(file);
      }
      catch (FileNotFoundException e) {
         System.out.println("File does not exist");
      }
   }
}
```

Outptut-
File does not exist

12.
IOException-

```
import java.io.*;
class Excercise12 {
public static void main(String args[])
   {
      FileInputStream f = null;
      f = new FileInputStream("abc.txt");
      int i;
      while ((i = f.read()) != -1) {
         System.out.print((char)i);
      }
      f.close();
   }
}
```

Output-
error: unreported exception IOException; must be caught or declared to be thrown

13.
NoSuchFieldException-

```
import java.lang.reflect.*;
public class Excercise13
{
   public static void main(String args[])
   {
      Excercise13  obj = new Excercise13();
      Class class1 = obj.getClass();
      System.out.println("Field : ");
      try
     {
```

```
        Field Flds = class1.getField("str");
        System.out.println(" Field found: " + Flds.toString());
      }
      catch(NoSuchFieldException e)
     {
       System.out.println(e.toString());
     }
    }
}
```

Output-
Field :
java.lang.NoSuchFieldException: str

14.
NoSuchMethodException -

```
public class Excercise14 {
public Demo()
   {
      Class i;
      try {
         i = Class.forName("java.lang.String");
         try {
            Class[] p = new Class[5];
         }
         catch (SecurityException e) {
            e.printStackTrace();
         }
         catch (NoSuchMethodException e) {
            e.printStackTrace();
         }
      }
      catch (ClassNotFoundException e) {
         e.printStackTrace();
      }
   }

public static void main(String[] args)
   {
      Demo();
   }
}
```

Output-
error: exception NoSuchMethodException is never thrown in body of corresponding try statement

15.
NullPointerException-

```
public class Excercise15 {

public static void main(String[] args)
   {
      try {
         String a = null;
         System.out.println(a.charAt(0));
      }
      catch (NullPointerException e) {
         System.out.println("NullPointerException occurs");
      }
   }
}
```

Output-
NullPointerException occurs

16.
NumberFormatException-

```
public class Excercise16{

public static void main(String[] args)
   {
      try {
         int num = Integer.parseInt("abc");

         System.out.println(num);
      }
      catch (NumberFormatException e) {
         System.out.println("Number format exception occurs");
      }
   }
}
```

Output-
Number format exception occurs

17.
StringIndexOutOfBoundsException-

```
public class Excercise17 {

public static void main(String[] args)
   {
      try {
         String a = "Hello World!";
         char c = a.charAt(22);
         System.out.println(c);
      }
```

```
        catch (StringIndexOutOfBoundsException e) {
            System.out.println("StringIndexOutOfBoundsException occurs");
        }
    }
}
```

Output-
StringIndexOutOfBoundsException occurs