Nepathya College
Tilottama-5, Rupandehi

Object Oriented Programming in Java
Lab 6

Objective: To learn concepts of Handling Errors/ Exceptions and understand to use exceptions: try and catch, throwing and re-throwing, throw, throws, cleaning up with finally.
Descriptions:

- **Refer theory from the slides.**

Program
Note: The program should be well formatted i.e., proper use of indentation, comment, description of program and functions etc.

1. Run the following program and complete the program to demonstrate the concept of Exception Handling or run time exception.

```
public class HandlingException{
        public static void main(String[] args){
                int d, a;
                        d = 0;
                        a = 34 / d;
        }
}
```

Output

```
divide by zero
after catch statement programs flows
```

2. Find the problem and complete the following program.

```
public class HandlingException {
        public static void main(String[] args) {
                int[] num = {1, 2, 3, 4};
                System.out.println(num[5]);
        }
}
```

Output

```
Invalid array indexs!!!java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for leng
Statement after exception generating statement
```

Prepared by: Shiva Bhattarai

3. Run the program that demonstrate the multiple catch blocks.

```
public class ExceptionDemo{
	public static void main(String[] args) {
		try {
			int a = args.length;
			System.out.println("a = "+a);
			int [] c = {1};
			c[42] = 99;
		}catch(ArithmeticException e) {
			System.out.println("Divide by 0: " + e);
		}catch(ArrayIndexOutOfBoundsException e) {
			System.out.println("Array index oob: " + e);
		}
		System.out.println("After try/catch blocks.");
	}
}
```

4. Run the program and Demonstrate the keyword throw. Also answer why "static" keyword is written in validate function.

```
public class ExceptionDemo{
	static void validate (int age) {
		try {
			if (age < 18)
				throw new ArithmeticException("Not Valid");
			else
				System.out.println("Can vote");
		}catch(Exception e) {
			System.out.println(e);
		}
	}
	public static void main(String[] args) {
		validate(12);
	}
}
```

5. Run and Demonstrate the keyword throws.

```
public class ExceptionDemo{
	int division (int a, int b) throws ArithmeticException{
		int t = a / b;
		return t;
	}
	public static void main(String[] args) {
		try {
			ExceptionDemo a = new ExceptionDemo();
			a.division(10, 0);
		}catch(ArithmeticException e) {
			System.out.println(e);
		}
	}
}
```