Vanier College
Computer Science Department

Programming 2

LAB 7

Q1: What output is produced by the following code?

```
int waitTime = 46;
try
{
     System.out.println("Try block entered.");
     if (waitTime > 30)
          throw new Exception("Over 30.");
     else if (waitTime < 30)
          throw new Exception("Under 30.");
     else
          System.out.println("No exception.");
     System.out.println("Leaving try block.");
}
catch(Exception thrownObject)
{
     System.out.println(thrownObject.getMessage());
}
System.out.println("After catch block");
```

Q2: Is the following legal?

```
Exception exceptionObject = new Exception("Oops!");
```

Q3: Is the following legal?

```
Exception exceptionObject = new Exception("Oops!");
throw exceptionObject;
```

Q4: Write a Java statement that throws a new exception with the String: File Not Found.

Q5: Use a catch block to display the exception thrown in Q4.

Q6: Define an exception class called FileNotFoundException. The class should have a constructor with no parameters. If an exception is thrown with this zero-argument constructor, getMessage should return "File Not Found!" The class should also have a constructor with a single parameter of type String. If an exception is thrown with this constructor, then getMessage returns the value that was used as an argument to the constructor.

Q7: Write a program that calculates the average of N integers. The program should prompt the user to enter the value for N and then afterward must enter all N numbers. If the user enters a nonpositive value for N, then an exception should be thrown (and caught) with the message "N must be positive." If there is any exception as the user is entering the N numbers, an error message should be displayed, and the user prompted to enter the number again.

Q8: Here is a snippet of code that inputs two integers and divides them:

```
Scanner scan = new Scanner(System.in);

int n1, n2;

double r;

n1 = scan.nextInt();

n2 = scan.nextInt();

r = ( double) n1 / n2;
```

Place this code into a try-catch block with multiple catches so that different error messages are printed if we attempt to divide by zero or if the user enters textual data instead of integers (java.util.InputMismatchException). If either of these conditions occurs, then the program should loop back and let the user enter new data.