ASSIGMENT JAVA DAY7

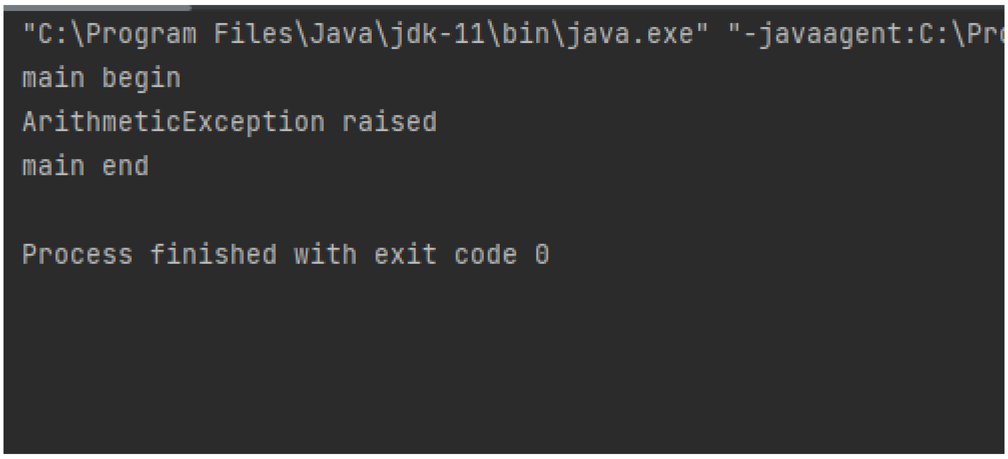
Harshit Kushmakar| 16896

1. Write a program to demonstrate the use of try, catch, finally throw and throws keywords and demonstrate the following points in the program.

a) Multiple catch blocks.

```
package assignment7;
import java.util.Arrays;

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



            System.out.println("main begin");
            try {
                int a[] = new int[5];
                a[10] = 5 / 0;
                // Here try followed by 2 catch blocks
            } catch (ArithmeticException e) {
                // this catch is to handle ArithmeticException
                System.out.println("ArithmeticException raised");
            } catch (ArrayIndexOutOfBoundsException e) {
                // this catch is to handle ArrayIndexOutOfBoundsException
                System.out.println("ArrayIndexOutOfBoundsException
raised");
            }
            System.out.println("main end");

        }
    }
```

OUTPUT:

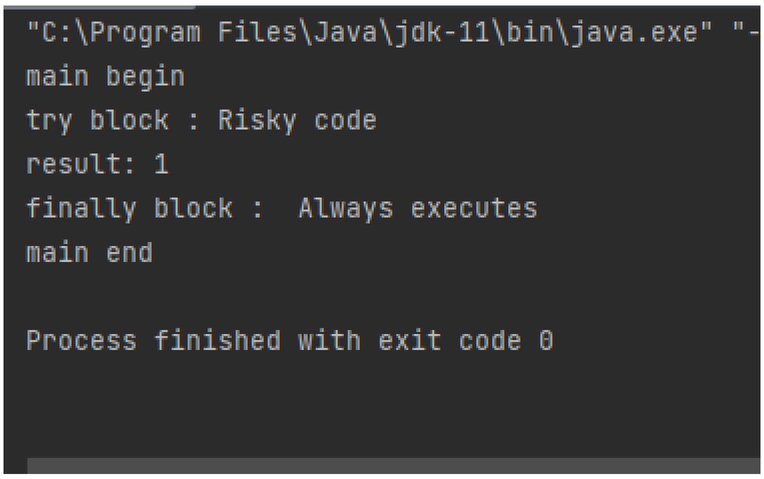


```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Pr
main begin
ArithmeticException raised
main end

Process finished with exit code 0
```

b) try-catch-finally combination.

```
package assignment7;
import java.util.Arrays;

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

        System.out.println("main begin");
            try {
                System.out.println("try block : Risky code");
                int result = 5 / 5;
                System.out.println("result: " + result);
            } catch (ArithmeticException e) {
                System.out.println("catch-block : Handling
ArithmeticException");
            } finally {
                // finally executes as exception not raised
                System.out.println("finally block :  Always executes");
            }

            System.out.println("main end");
        }
    }
```

OUTPUT:



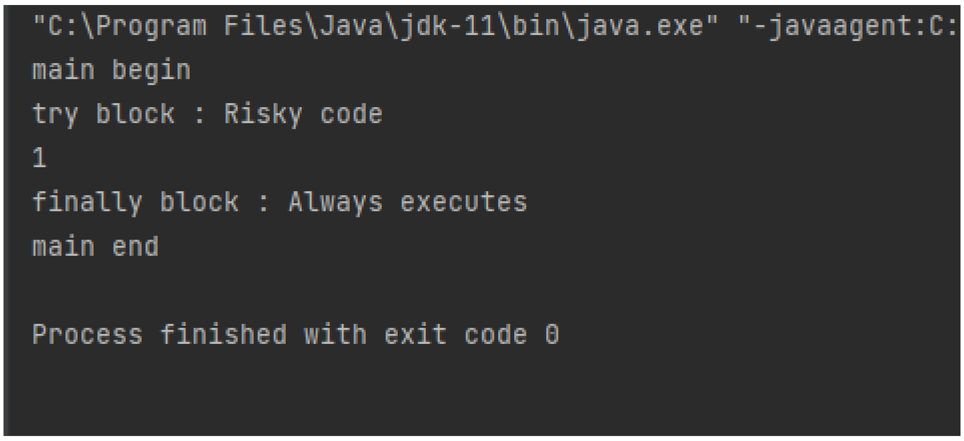
```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-
main begin
try block : Risky code
result: 1
finally block :  Always executes
main end

Process finished with exit code 0
```

c) try-finally combination.

```
package assignment7;

public class TryFinally {

    public static void main(String[] args) {

        System.out.println("main begin");
        try {
            System.out.println("try block : Risky code");
            System.out.println(5 / 5);
        } finally {
```

```java
            // finally executes as exception not raised
            System.out.println("finally block : Always executes");
        }
        System.out.println("main end");
    }
}
```

OUTPUT:

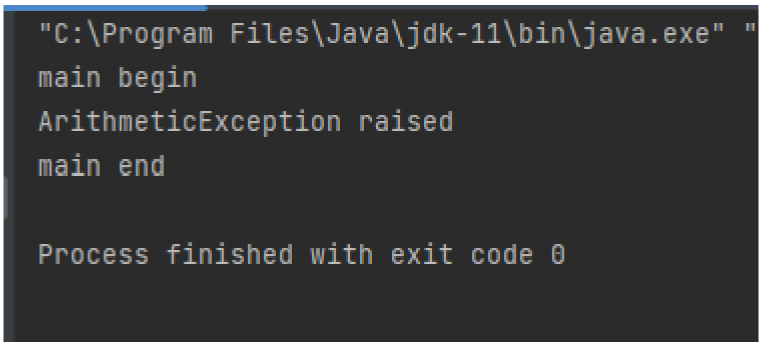


```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:
main begin
try block : Risky code
1
finally block : Always executes
main end

Process finished with exit code 0
```

d) Exception propagation among many methods.

```java
package assignment7;

public class ExceptionProp {
    public void method1() {
        int result = 5 / 0;
        // exception propagated to method2()
    }

    public void method2() {
        method1();
        // exception propagated to method3()
    }

    public void method3() {
        try {
            method2();
            // if not handled here exception propagated to main() method
        } catch (ArithmeticException e) {
            System.out.println("exception handled");
        }
    }

    public static void main(String args[]) {
        // // if not handled in main() method exception propagated to
        // defaultExceptionHandler
        System.out.println("main() begin");
        ExceptionProp obj = new ExceptionProp();
        obj.method3();
        System.out.println("main() end");
    }
}
```

OUTPUT:

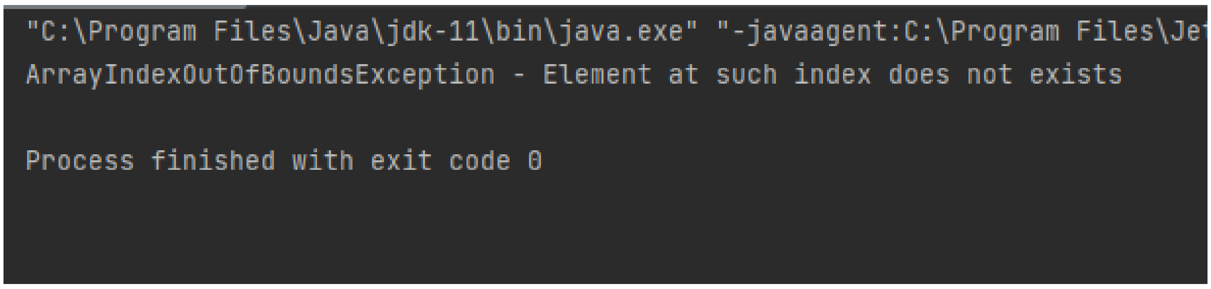


```
"C:\Program Files\Java\jdk-11\bin\java.exe" "
main begin
ArithmeticException raised
main end

Process finished with exit code 0
```

e) Nested try blocks.

```
package assignment7;

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

        try {

            // initializing array
            int a[] = {1, 2, 3, 4, 5};

            // trying to print element at index 5
            System.out.println(a[5]);

            // try-block2 inside another try block
            try {

                // performing division by zero
                int x = a[2] / 0;
            } catch (ArithmeticException e2) {
                System.out.println("division by zero is not possible");
            }
        } catch (ArrayIndexOutOfBoundsException e1) {
            System.out.println("ArrayIndexOutOfBoundsException - Element at
such index does not exists");
        }
    }
// end of main method

}
```

OUTPUT:



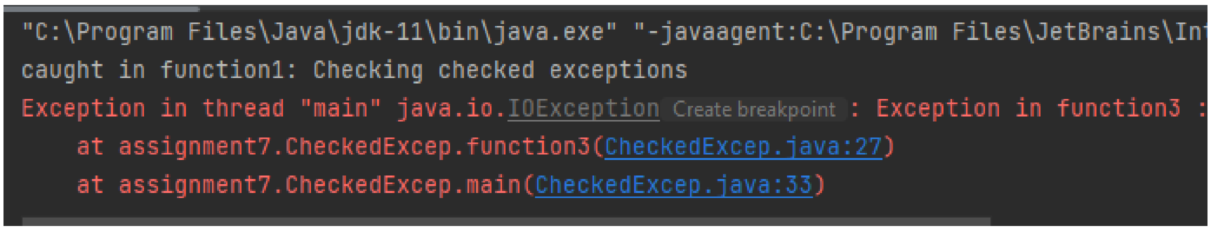
```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Program Files\Je
ArrayIndexOutOfBoundsException - Element at such index does not exists

Process finished with exit code 0
```

2. Write a program to throw a checked exception explicitly using 'throw' keyword and

a) Handle the exception in same method.

```java
package assignment7;

import java.io.IOException;

public class CheckedExcep {
    public static void function1() throws IOException {
        boolean a = true;
        try {

            if (a) {
                throw new IOException("Checking checked exceptions");
            }
        } catch (IOException e) {
            System.out.println("caught in function1: " +e.getMessage());
        }
    }
    public static void function2() {
    try{
        function1();  // // handling exception in other method

    }
    catch (Exception e){
        System.out.println("caught in function2:" +e.getMessage());
    }
    }
    public static void function3() throws IOException{
        throw new IOException("Exception in function3 : ");

    }

    public static void main(String[] args) throws Exception{
        function2();
        function3();
    }
}
```

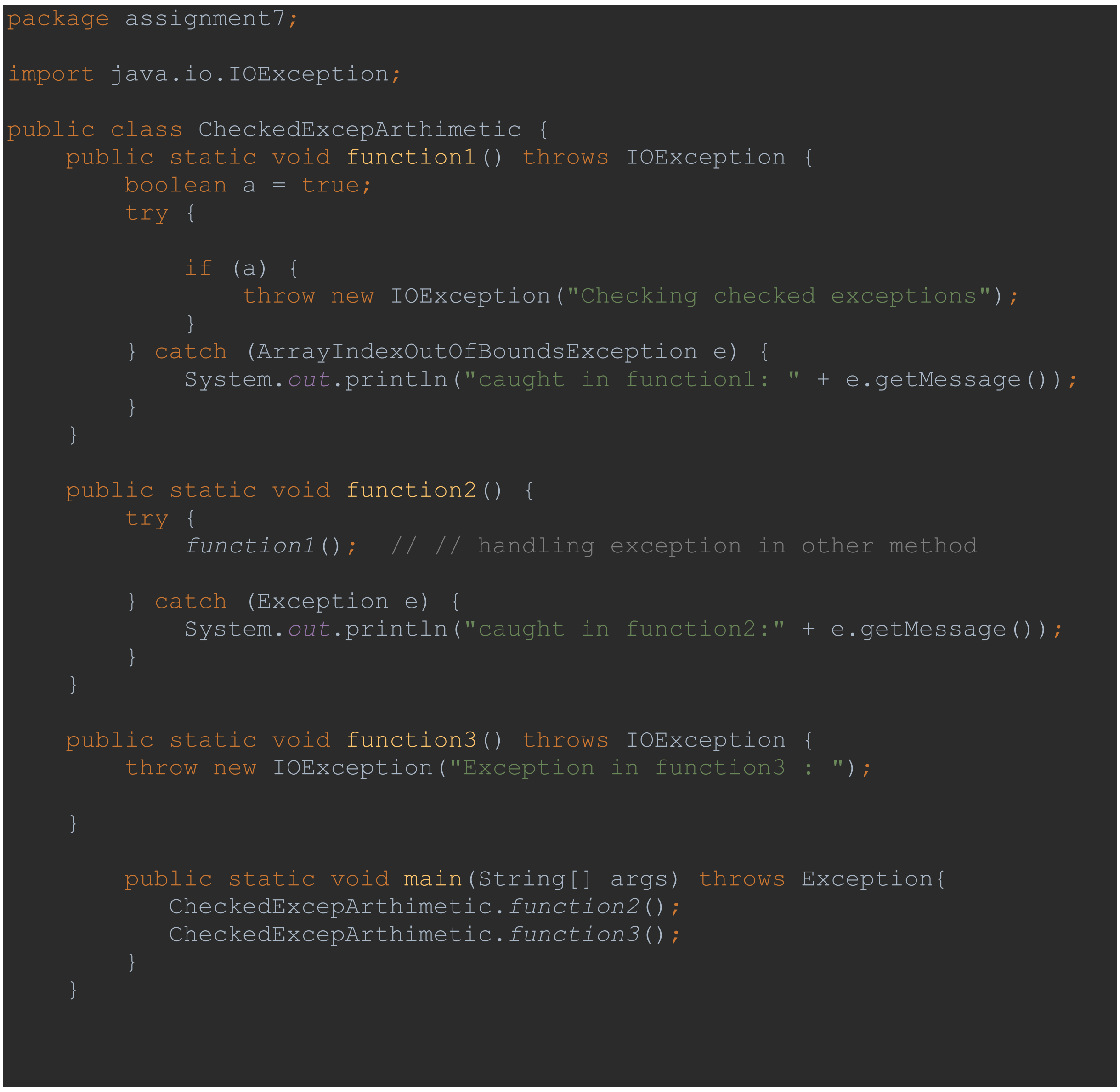
OUTPUT:



```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\Int
caught in function1: Checking checked exceptions
Exception in thread "main" java.io.IOException Create breakpoint : Exception in function3 :
    at assignment7.CheckedExcep.function3(CheckedExcep.java:27)
    at assignment7.CheckedExcep.main(CheckedExcep.java:33)
```

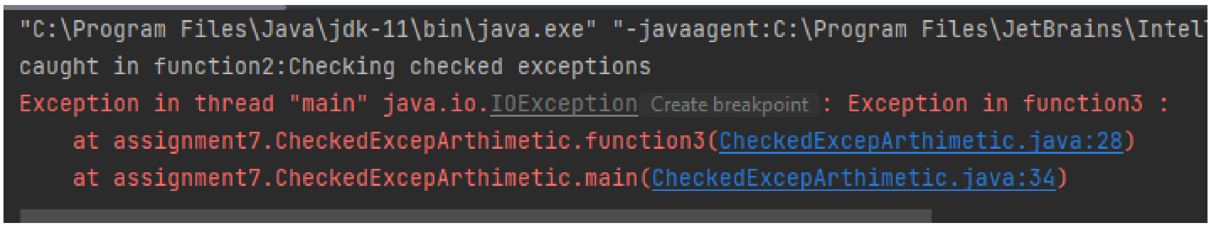
b) use throws clause and handle the exception in some other method (calling method) .

c) Don't either handle or use the throws clause.



```java
package assignment7;

import java.io.IOException;

public class CheckedExcepArthimetic {
    public static void function1() throws IOException {
        boolean a = true;
        try {

            if (a) {
                throw new IOException("Checking checked exceptions");
            }
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("caught in function1: " + e.getMessage());
        }
    }

    public static void function2() {
        try {
            function1();  // // handling exception in other method

        } catch (Exception e) {
            System.out.println("caught in function2:" + e.getMessage());
        }
    }

    public static void function3() throws IOException {
        throw new IOException("Exception in function3 : ");

    }

        public static void main(String[] args) throws Exception{
           CheckedExcepArthimetic.function2();
           CheckedExcepArthimetic.function3();
        }
    }
```

OUTPUT:



```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\Intel
caught in function2:Checking checked exceptions
Exception in thread "main" java.io.IOException Create breakpoint : Exception in function3 :
    at assignment7.CheckedExcepArthimetic.function3(CheckedExcepArthimetic.java:28)
    at assignment7.CheckedExcepArthimetic.main(CheckedExcepArthimetic.java:34)
```

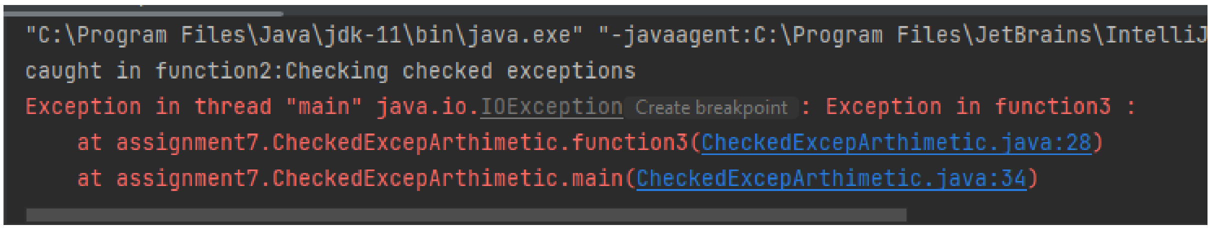
3. Write a program to throw an unchecked exception explicitly using 'throw' keyword and

a) Handle the exception in same method.

b) use throws clause and handle the exception in some other method (calling method)

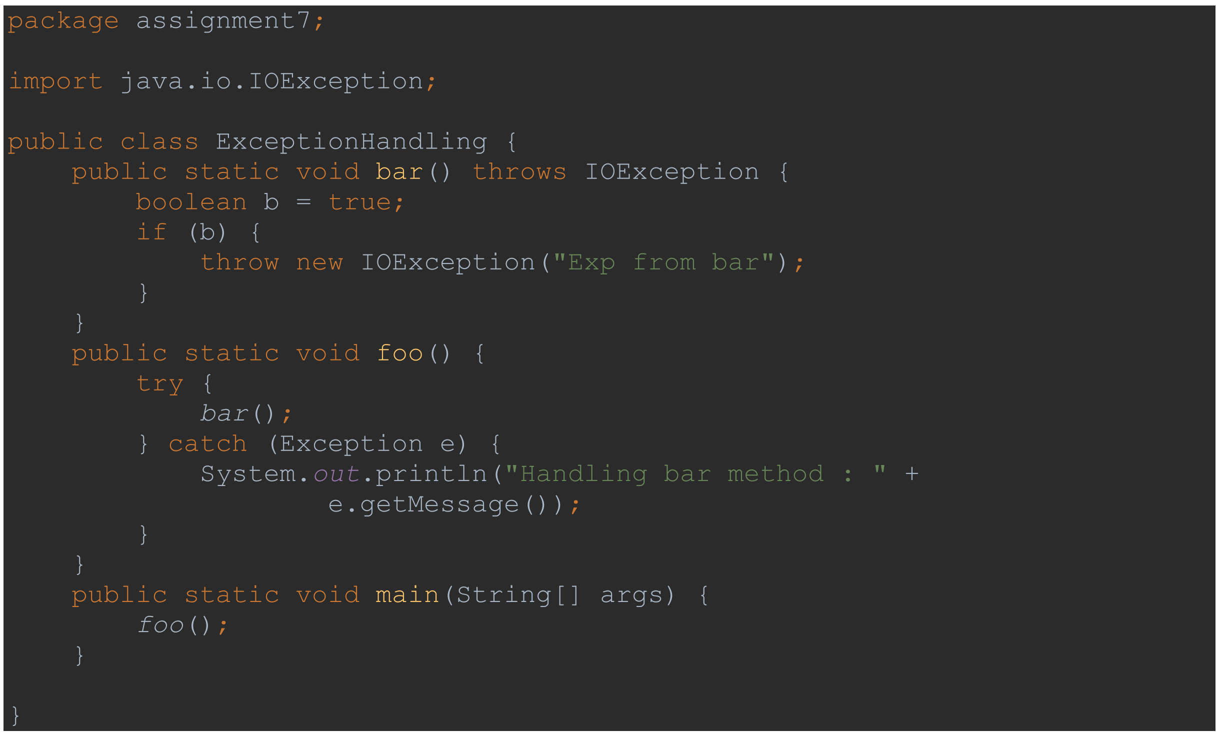
c) Don't either handle or use the throws clause.

```
package assignment7;

public class UncheckedExcep {
    public static void function1() throws ArithmeticException {
        System.out.println(12 / 0);
    }
    public static void function2() {
        try {
            function1();
        } catch (ArithmeticException e) { // Handling expression in other
method(3(b))
            System.out.println("Handling arithmetic exp in funcExp2 : " +
e.getMessage());
        }
    }
    public static void function3() {
        try { //  Handling expression in same method
            int a = 12 / 0;
        } catch (ArithmeticException e) {
            System.out.println("In funcExp3 catch : " +
                    e.getMessage());
        }
    }
    public static void main(String[] args) {
         UncheckedExcep.function2();
        UncheckedExcep.function3();
    }
}
```



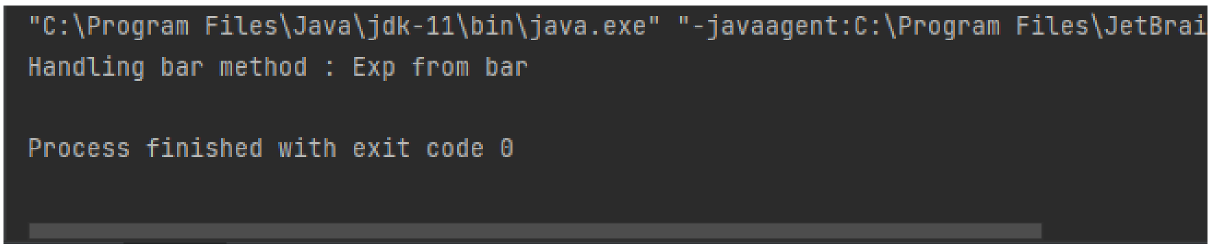
```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ
caught in function2:Checking checked exceptions
Exception in thread "main" java.io.IOException Create breakpoint : Exception in function3 :
    at assignment7.CheckedExcepArthimetic.function3(CheckedExcepArthimetic.java:28)
    at assignment7.CheckedExcepArthimetic.main(CheckedExcepArthimetic.java:34)
```

4. Write a program in which main method calls the foo method which calls the bar method. Bar method can throw a checked exception. Use throws for throwing the exception from bar. Don't handle exception in bar using try catch. Let the calling function handle the same.



```java
package assignment7;

import java.io.IOException;

public class ExceptionHandling {
    public static void bar() throws IOException {
        boolean b = true;
        if (b) {
            throw new IOException("Exp from bar");
        }
    }
    public static void foo() {
        try {
            bar();
        } catch (Exception e) {
            System.out.println("Handling bar method : " +
                    e.getMessage());
        }
    }
    public static void main(String[] args) {
        foo();
    }

}
```

OUTPUT:



```
"C:\Program Files\Java\jdk-11\bin\java.exe" "-javaagent:C:\Program Files\JetBrai
Handling bar method : Exp from bar

Process finished with exit code 0
```