Programs Demonstrating Exceptions

1. **NullPointerExceptionThrown.java**

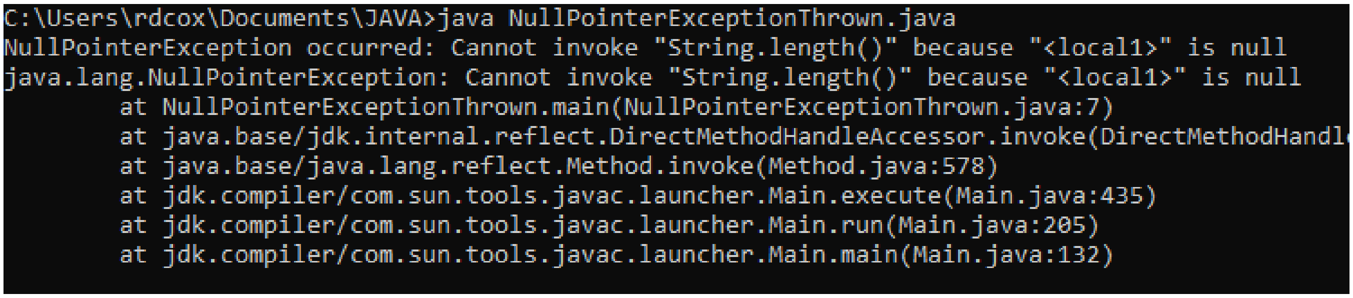
   Program that intentionally throws a NullPointerException.

   This exception occurs when you try to access or manipulate an object that has a null value.

//Code

```
public class NullPointerExceptionThrown {
   public static void main(String[] args) {
      String str = null; // Initializing a String variable as null

      try {
         // Attempting to get the length of a null String
         int length = str.length(); // This will throw NullPointerException
      } catch (NullPointerException e) {
         // Handling the NullPointerException
         System.out.println("NullPointerException occurred: " + e.getMessage());
         e.printStackTrace(); // Displaying stack trace for more information
      }
   }
}
```

//output



```
C:\Users\rdcox\Documents\JAVA>java NullPointerExceptionThrown.java
NullPointerException occurred: Cannot invoke "String.length()" because "<local1>" is null
java.lang.NullPointerException: Cannot invoke "String.length()" because "<local1>" is null
        at NullPointerExceptionThrown.main(NullPointerExceptionThrown.java:7)
        at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandl
        at java.base/java.lang.reflect.Method.invoke(Method.java:578)
        at jdk.compiler/com.sun.tools.javac.launcher.Main.execute(Main.java:435)
        at jdk.compiler/com.sun.tools.javac.launcher.Main.run(Main.java:205)
        at jdk.compiler/com.sun.tools.javac.launcher.Main.main(Main.java:132)
```

2. ArrayIndexOutOfBoundsExceptionThrown.java

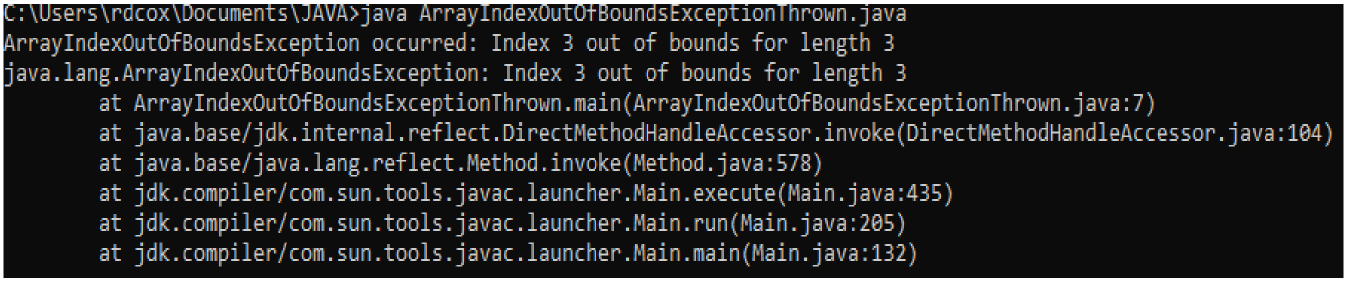
Program that intentionally throws an ArrayIndexOutOfBoundsException.

This exception occurs when you try to access an array element with an index that is out of the array's bounds.

//code

```
public class ArrayIndexOutOfBoundsExceptionThrown {
  public static void main(String[] args) {
    int[] arr = {1, 2, 3}; // Initializing an array

    try {
      // Accessing an index outside the array bounds
      int element = arr[3]; // This will throw ArrayIndexOutOfBoundsException
    } catch (ArrayIndexOutOfBoundsException e) {
      // Handling the ArrayIndexOutOfBoundsException
      System.out.println("ArrayIndexOutOfBoundsException occurred: " + e.getMessage());
      e.printStackTrace(); // Displaying stack trace for more information
    }
  }
}
```

//output



```
C:\Users\rdcox\Documents\JAVA>java ArrayIndexOutOfBoundsExceptionThrown.java
ArrayIndexOutOfBoundsException occurred: Index 3 out of bounds for length 3
java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3
        at ArrayIndexOutOfBoundsExceptionThrown.main(ArrayIndexOutOfBoundsExceptionThrown.java:7)
        at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:104)
        at java.base/java.lang.reflect.Method.invoke(Method.java:578)
        at jdk.compiler/com.sun.tools.javac.launcher.Main.execute(Main.java:435)
        at jdk.compiler/com.sun.tools.javac.launcher.Main.run(Main.java:205)
        at jdk.compiler/com.sun.tools.javac.launcher.Main.main(Main.java:132)
```

3. ClassCastExceptionThrown.java

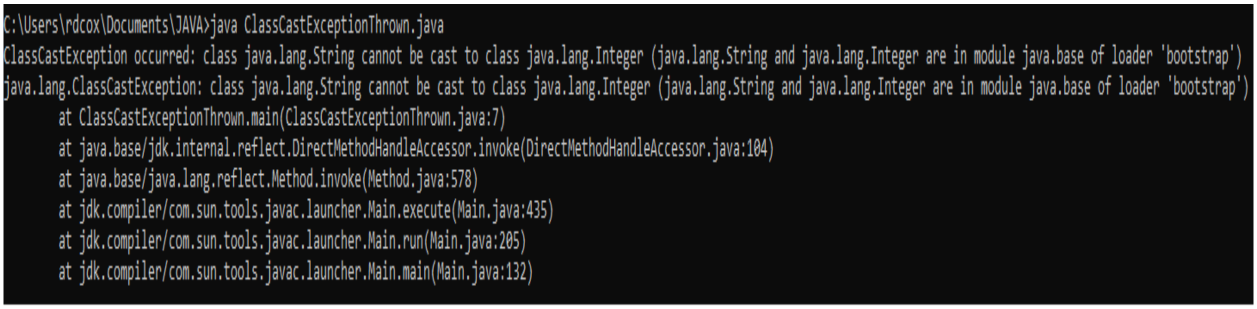
Program that intentionally throws a ClassCastException.

This exception occurs when you try to cast an object to a type that it's not compatible with.

//code

```
public class ClassCastExceptionThrown {
  public static void main(String[] args) {
    try {
      Object someObject = "Hello"; // Creating an Object and assigning a String to it

      // Attempting to cast the Object to an incompatible type (Integer in this case)
      Integer num = (Integer) someObject; // This will throw ClassCastException
    } catch (ClassCastException e) {
      // Handling the ClassCastException
      System.out.println("ClassCastException occurred: " + e.getMessage());
      e.printStackTrace(); // Displaying stack trace for more information
    }
  }
}
```

//output



```
C:\Users\rdcox\Documents\JAVA>java ClassCastExceptionThrown.java
ClassCastException occurred: class java.lang.String cannot be cast to class java.lang.Integer (java.lang.String and java.lang.Integer are in module java.base of loader 'bootstrap')
java.lang.ClassCastException: class java.lang.String cannot be cast to class java.lang.Integer (java.lang.String and java.lang.Integer are in module java.base of loader 'bootstrap')
        at ClassCastExceptionThrown.main(ClassCastExceptionThrown.java:7)
        at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:104)
        at java.base/java.lang.reflect.Method.invoke(Method.java:578)
        at jdk.compiler/com.sun.tools.javac.launcher.Main.execute(Main.java:435)
        at jdk.compiler/com.sun.tools.javac.launcher.Main.run(Main.java:205)
        at jdk.compiler/com.sun.tools.javac.launcher.Main.main(Main.java:132)
```

4. IllegalArgumentExceptionThrown.java

Program that intentionally throws an IllegalArgumentException.

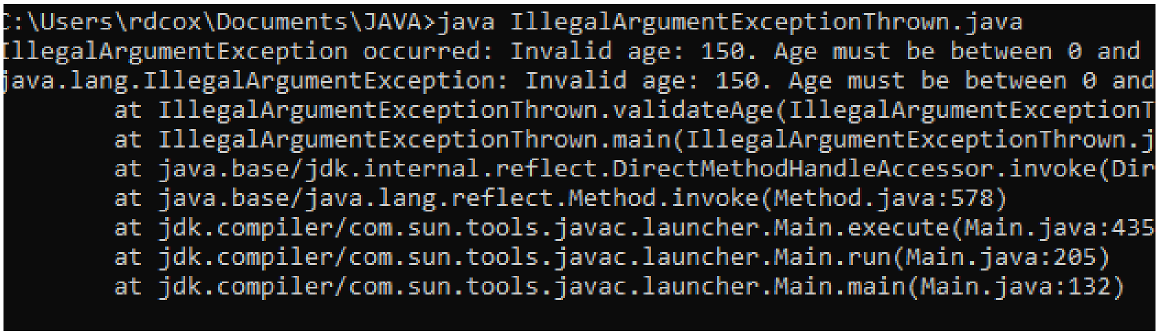
This exception occurs when an illegal or inappropriate argument is passed to a method.

//code

```
public class IllegalArgumentExceptionThrown {

  public static void validateAge(int age) {

    if (age < 0 || age > 120) {

      throw new IllegalArgumentException("Invalid age: " + age + ". Age must be between 0 and 120.");

    }

    System.out.println("Valid age: " + age);

  }


  public static void main(String[] args) {

    try {

      validateAge(150); // Passing an invalid age

    } catch (IllegalArgumentException) {

      // Handling the IllegalArgumentException

      System.out.println("IllegalArgumentException occurred: " + e.getMessage());

      e.printStackTrace(); // Displaying stack trace for more information
```

```
        }

    }
```

//output



```
C:\Users\rdcox\Documents\JAVA>java IllegalArgumentExceptionThrown.java
IllegalArgumentException occurred: Invalid age: 150. Age must be between 0 and
java.lang.IllegalArgumentException: Invalid age: 150. Age must be between 0 and
        at IllegalArgumentExceptionThrown.validateAge(IllegalArgumentExceptionT
        at IllegalArgumentExceptionThrown.main(IllegalArgumentExceptionThrown.j
        at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(Dir
        at java.base/java.lang.reflect.Method.invoke(Method.java:578)
        at jdk.compiler/com.sun.tools.javac.launcher.Main.execute(Main.java:435
        at jdk.compiler/com.sun.tools.javac.launcher.Main.run(Main.java:205)
        at jdk.compiler/com.sun.tools.javac.launcher.Main.main(Main.java:132)
```

Programs Catching Exceptions

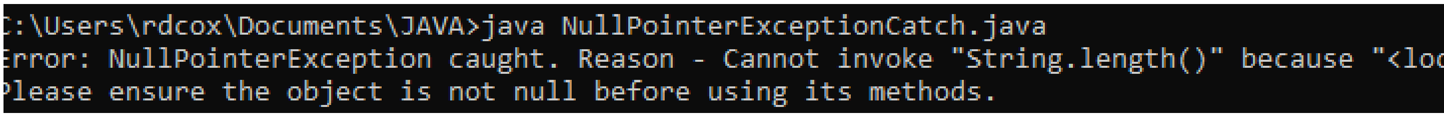
1. **NullPointerExceptionCatch.java**

    Program that catches and handles a NullPointerException, displaying an error message.

//code (same as 1 above, just with error message.

```
 public class NullPointerExceptionCatch {

    public static void main(String[] args) {

        String str = null; // Initializing a String variable as null



        try {

            // Attempting to get the length of a null String

            int length = str.length(); // This will throw NullPointerException

        } catch (NullPointerException e) {

            // Handling the NullPointerException

            System.out.println("Error: NullPointerException caught. Reason - " + e.getMessage());

            System.out.println("Please ensure the object is not null before using its methods.");
        }
    }
}
```

//output



```
C:\Users\rdcox\Documents\JAVA>java NullPointerExceptionCatch.java
Error: NullPointerException caught. Reason - Cannot invoke "String.length()" because "<loc
Please ensure the object is not null before using its methods.
```

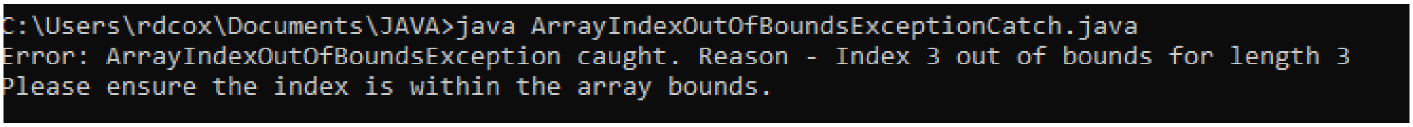
2. **ArrayIndexOutOfBoundsExceptionCatch.java**

Program that catches and handles an ArrayIndexOutOfBoundsException, displaying an error message.

//code

```
public class ArrayIndexOutOfBoundsExceptionCatch {
    public static void main(String[] args) {
        int[] numbers = { 1, 2, 3 };

        try {
            // Accessing an index outside the array bounds
            int element = numbers[3]; // This will throw ArrayIndexOutOfBoundsException
        } catch (ArrayIndexOutOfBoundsException e) {
            // Handling the ArrayIndexOutOfBoundsException
            System.out.println("Error: ArrayIndexOutOfBoundsException caught. Reason - " + e.getMessage());
            System.out.println("Please ensure the index is within the array bounds.");
        }
    }
}
```

//output



```
C:\Users\rdcox\Documents\JAVA>java ArrayIndexOutOfBoundsExceptionCatch.java
Error: ArrayIndexOutOfBoundsException caught. Reason - Index 3 out of bounds for length 3
Please ensure the index is within the array bounds.
```

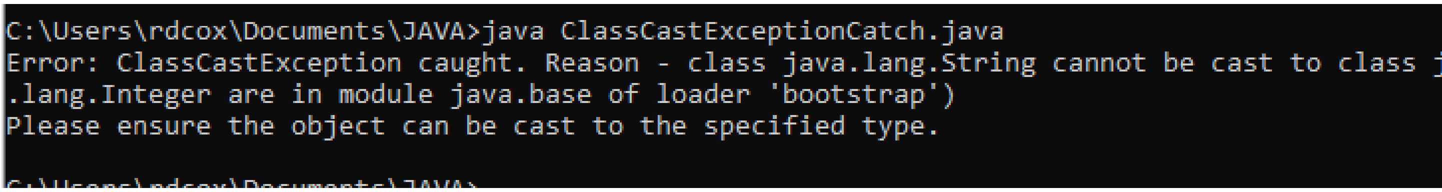
3. ClassCastExceptionCatch.java

Program that catches and handles a ClassCastException, displaying an error message.

//code

```
public class ClassCastExceptionCatch {
    public static void main(String[] args) {
        try {
            Object someObject = "Hello"; // Creating an Object and assigning a String to it

            // Attempting to cast the Object to an incompatible type (Integer in this case)
            Integer num = (Integer) someObject; // This will throw ClassCastException
        } catch (ClassCastException e) {
            // Handling the ClassCastException
            System.out.println("Error: ClassCastException caught. Reason - " + e.getMessage());
            System.out.println("Please ensure the object can be cast to the specified type.");
        }
    }
}
```

//output



```
C:\Users\rdcox\Documents\JAVA>java ClassCastExceptionCatch.java
Error: ClassCastException caught. Reason - class java.lang.String cannot be cast to class j
.lang.Integer are in module java.base of loader 'bootstrap')
Please ensure the object can be cast to the specified type.

C:\Users\rdcox\Documents\JAVA>
```

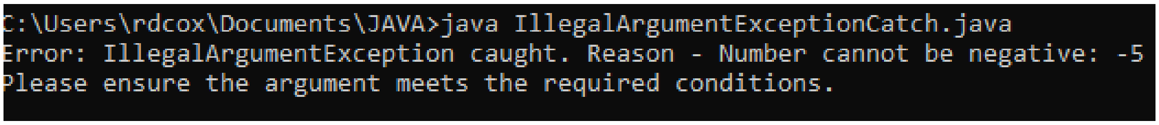
4. **IllegalArgumentExceptionCatch.java**

   Program that catches and handles an IllegalArgumentException, displaying an error message.

//code

```
public class IllegalArgumentExceptionCatch {
  public static void main(String[] args) {
    try {
      int number = -5; // Initializing a number (int) with an invalid value

      if (number < 0) {
        throw new IllegalArgumentException("Number cannot be negative: " + number);
      }
    } catch (IllegalArgumentException e) {
      // Handling the IllegalArgumentException
      System.out.println("Error: IllegalArgumentException caught. Reason - " + e.getMessage());
      System.out.println("Please ensure the argument meets the required conditions.");
    }
  }
}
```

//output



```
C:\Users\rdcox\Documents\JAVA>java IllegalArgumentExceptionCatch.java
Error: IllegalArgumentException caught. Reason - Number cannot be negative: -5
Please ensure the argument meets the required conditions.
```