

# How to fix ArrayStoreException in Java?

## ArrayStoreException in Java

---

In this tutorial, we will learn how to fix `java.lang.ArrayStoreException` that occurs when an object of a different type is made to be stored in an array of a type.

`java.lang.ArrayStoreException` occurs when we try to store an object of a type in an array of objects of different type. Usually, one would come across “`java.lang.ArrayStoreException: java.lang.Integer`” which occurs when an attempt is made to store an integer in an array of different type like array of String or array of float, etc.

Lets see a sample program that could produce this exception.

### ArrayStoreExceptionExample.java

```
package com.tutorialkart.java;

/**
 * @author tutorialkart.com
 */
public class ArrayStoreExceptionExample {
    public static void main(String[] args) {
        Object[] names = new Float[2];
        names[1] = new Integer(2);
    }
}
```

And when this program is run, following would be the result.

### Output

```
Exception in thread "main" java.lang.ArrayStoreException: java.lang.Integer
    at com.tutorialkart.java.ArrayStoreExceptionExample.main(ArrayStoreExceptionExample.
```

Let us see the first line of stack trace, “`java.lang.ArrayStoreException: java.lang.Integer` in detail.

- `java.lang.ArrayStoreException`: Exception thrown by java language when we try to store an object of `java.lang.Integer` in an array of `java.lang.Float`.
- `java.lang.Integer`: Integer is the type of object that has been tried to store in array of different type (here its Float)

## How to Handle ArrayStoreException?

---

Lets handle the ArrayStoreException using try-catch.

- Surround the statements that could throw ArrayStoreException with try-catch block.
- Catch ArrayStoreException.
- Take necessary action for the further course of your program, as you are handling the exception and the execution doesn't abort.

### ArrayStoreExceptionExample.java

```
package com.tutorialkart.java;

/**
 * @author tutorialkart.com
 */
public class ArrayStoreExceptionExample {
    public static void main(String[] args) {
        Object[] names = new Float[2];
        try {
            names[1] = new Integer(2);
        } catch (ArrayStoreException e) {
            e.printStackTrace();
            System.out.println("The stack trace has been printed just for logging.");
            System.out.println("ArrayStoreException is handled.");
        }
        System.out.println("Continuing with the statements after try-catch block.");
    }
}
```

And when you run this java program, result would be as shown in the following.

### Output

```
java.lang.ArrayStoreException: java.lang.Integer
    at com.tutorialkart.java.ArrayStoreExceptionExample.main(ArrayStoreExceptionExample.
The stack trace has been printed just for logging.
ArrayStoreException is handled.
Continuing with the statements after try-catch block.
```

When an exception occurs, the execution falls on to the catch block from the point of occurrence of exception. It executes the statement in catch block and continues with the statement present after the try-catch block. So, care has to be taken while handling the exceptions and deciding on the further course of action in your program.

## Conclusion

---

In this [Java Tutorial](#), we have learnt to fix java.lang.ArrayStoreException that occurs when an object of a

different type is made to be stored in an array of a type.

## Java Tutorial

- ◆ [Java Tutorial](#)
- ◆ [Java Introduction](#)
- ◆ [Java Installation](#)
- ◆ [IDEs for Java Development](#)
- ◆ [Java HelloWorld Program](#)
- ◆ [Java Program Structure](#)
- ◆ [Java Datatypes](#)
- ◆ [Java Variable Types](#)
- ◆ [Java Access Modifiers](#)
- ◆ [Java Operators](#)
- ◆ [Java Decision Making](#)
- ◆ [Java Loops](#)
- ◆ [Java Array](#)
- ◆ [Java OOPs](#)
- ◆ [Java String](#)
- ◆ [Java Exception Handling](#)
- ◆ [Java File Operations](#)
- ◆ [Java Date & Time](#)
- ◆ [Java MySQL](#)
- ◆ [Java Random](#)
- ◆ [Java Math](#)