Create BankAccount class which has private fields accountNumber, balance Add constructor to initialize the accountNumber and balance. Create methods deposit () and withdraw (), where: Deposit () adds to the balance. Withdraw () subtracts from the balance, but only if sufficient funds are available. In the main method, create an object of BankAccount, and demonstrate deposit and withdraw.

Create a java program which initializes an array with 3 elements and attempt to print the element at 3rd index. Write code and explain what happens when the program is executed.

Why does it result in an exception?

array out of bound exception will occur because array index start with 0.

3 elements means 0,1,2 we are accessing 3rd index which is not possible.

```
package exceptionEx;

public class ArrayIndexOBExample {
    public static void main(String[] args) {
        int[] numbers = {10,20,30};
try {
    //access the 3rd index element
    System.out.println("Element at 3rd index: " + numbers[3]);
}catch (ArrayIndexOutOfBoundsException e){
    //Handling the exception
    System.out.println(e.getMessage());
}
```

```
        System.out.println("Code to be executed after trying
to access 3rd element");
    }
}
```

Explanation:

Try block: the code which might throw an exception is placed inside the try block

Catch block: The ArrayIndexOutOfBoundException is caught and handled inside the catch block with proper message to user

After handling the exception, program continues to execute.

Method to print exception message:

1. printStackTrace():
   Prints detailed information about the exception and its origin.

2. getMessage():
   provides message associated with the exception

3. toString():
   provides the description of the exception, shows the class name and also the message.

NULLPOINTER Exception: TASK

Throw Keyword:
It is used to explicitly throw an exception in a method or block of code.

SYNTAX:

```
throw new ExceptionType(“Error Message”);
```

```
package throwExample;

public class ThrowCodeExample {
    public static void checkAge(int age){
        if (age < 18){
            throw new IllegalArgumentException("Age must be 18
or higher to vote!");
        }
    }

    public static void main(String[] args) {
        checkAge(13);
    }
}
```

#Throws Keyword
Declares exceptions that a method can throw to its caller.

```
package throwExample;

public class ThrowsCodeExample {
    public static void riskMethod() throws ArithmeticException
    {
        int result = 10/0;
```

```
        System.out.println("Result: "+result);
    }

    public static void main(String[] args) {
        try {
            riskMethod(); // Caller handles the exception
        }catch (ArithmeticException e){
            System.out.println("Exception Occured: "+
e.getMessage());
        }
    }
}
```

	Throw	Throws
Purpose	Used to explicitly throw an exception	Declares the exception a method might throw
Usage	Used in the method body	In the method signature
Number of exception	Only exception can be handled at a time	Can declare multiple exception(Comma seperation)

Custom Exceptions:*