WEEK-12

Question: Write a program that calls a method that throws an exception of type ArithmeticException in a for loop at an undesirable situation (such as divide by zero or taking square root of negative number). Catch the exception and display appropriate message. (Example of Unchecked Exception). Code:

```
class UncheckedExample {     static void riskyMethod(int a) {
      System.out.println(10 / a);
   }
   public static void main(String[] args) {
      for (int i = 2; i >= -2; i--) {
try {
            riskyMethod(i);
         } catch (ArithmeticException e) {
            System.out.println("Cannot divide by zero.");
         }
      }
   }
}
```

Output:

```
Output:
5
10
Cannot divide by zero.-10-5
```

Question: Write a program of your choice where a Checked Exception occurs at third function but handled at the first calling function. Use both ways of managing Checked Exception i.e. using try-catch block and throws keyword.

Code:

```
import java.io.*;  class CheckedEx {     static
void func3() throws IOException {
throw new IOException("File missing");
```

```
  }
  static void func2() throws IOException {
    func3();
  }
  static void func1() {
    try {
func2();
    } catch (IOException e) {
      System.out.println("Handled in func1: " + e.getMessage());
    }
  }
  public static void main(String[] args) {
func1();
  }
}
```

Output:

```
Output:
Handled in func1: File missing
```

Question: You are developing an online banking system where users can transfer money between accounts. If a user tries to withdraw more money than is available in their account, an InsufficientFundsException should be thrown.

Code:

```
class InsufficientFundsException extends Exception {
  InsufficientFundsException(String msg) { super(msg); }
}
class BankAccount {
double balance;
```

```
  BankAccount(double b){ balance=b; }     void withdraw(double amt) throws
InsufficientFundsException {        if(amt > balance) throw new
InsufficientFundsException("Not enough balance");
    balance -= amt;
  }
}
public class MainBank {     public static void
main(String[] args){        BankAccount b = new
BankAccount(1000);
    try {
      b.withdraw(1500);
    } catch (InsufficientFundsException e){
      System.out.println("Error: " + e.getMessage());
    }
  }
}
```

Output:

```
Output:
Error: Not enough balance
```

Question: Create a user-defined exception InvalidAgeException when the age of a person is below 18 years. Use this exception at appropriate place.

Code:

```
class InvalidAgeException extends Exception {
  InvalidAgeException(String msg){ super(msg); }
}
public class AgeCheck {     static void checkAge(int age)
throws InvalidAgeException {        if(age < 18) throw new
InvalidAgeException("Age must be 18+");
```

```
        System.out.println("Valid Age");
    }
    public static void main(String[] args) {
        try {
            checkAge(16);
        } catch (InvalidAgeException e){
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Output:

```
Output:
Error: Age must be 18
```