

Chapter 11 – Risky Behavior : exception handling

1. What type(s) of Exception(s) does the Java compiler ignore at compile-time and why does it ignore them? What is special about these type of Exceptions that they are not (and usually cannot) be caught at compile-time?

Answer :

2. Creating an entire class for possible “programming errors” or “Exceptions” seems like overkill. What is the advantage to representing “program errors” as object of type Exception?

Answer :

3. What type of code is typically placed in each part of a “try - catch” block?

Answer :

4. In class Chef below, method cookIngredients() makes a call to method “mixIngredients” that throws a MixingException. Modify the cookIngredients() method call the mixIngredients() method using a try-catch block. If a MixingException exception is thrown, print the statement:

“MixingException occurred”

```
public class Chef
{
    public void cookIngredients()
    {
        mixIngredients();    // throws MixingException
    }

} // end class Chef
```

Answer :

5. The method `cookIngredients()` does not necessarily need to call the `mixIngredients()` method using a try-catch block. It may instead “duck” the exception. Modify class `Chef` to “duck” the `MixingException` (forcing the code that calls the `cookIngredients()` method to deal with the `MixingException`).

Answer :

6. Examine the `ChefDriver` class below:

```
public class ChefDriver
{
    public static void main(String[] args)
    {
        Chef emeril = new Chef();

        emeril.cookIngredients();
    }
}
```

If the method `cookIngredients()` throws the exception `MixingException` (as in the question above), what will happen when the code “`emeril.cookIngredients()`” is executed?

Answer :

7. The Throwable class defines two important methods, getMessage() and printStackTrace(). Which of the following classes inherit these methods: IOException, InterruptedException, RuntimeException?

Answer :

8. Search the Java API the NullPointerException class. List three circumstances under which this exception be thrown. Give a Java code example that would throw this exception under each of these circumstances.

Answer :

9. Search the Java API the ClassCastException class. Under what circumstance will this exception be thrown? Give a Java code example that would throw this exception.

Answer :

10. Which of the following do you think are also RuntimeExceptions? Reference the Java API to check which of the following exceptions are RuntimeExceptions.

- a. IndexOutOfBoundsException**
- b. NoSuchMethodException**
- c. NegativeArraySizeException**
- d. StringIndexOutOfBoundsException**
- e. IllegalAccessException**

Answer :

11. Write the try-catch-finally block in class `PaycheckProcessingInc` that would successfully call the `Mainframe` method `processClients(server s)`. If the method is called successfully, the try block should print the statement “Connection successful”. If an exception is thrown, the catch block should print the statement “Server is down, please try again later”. A finally block should print the statement “Thank you for your business”.

```
public class PaycheckProcessingInc
{
    Mainframe.processClients(server s);
}

public class Mainframe
{
    public static void processClients(server s) throws
        NoServerException
    {
    }
} // end class Mainframe
```

Answer:

12. The try-catch block looks like an if-else statement. Why aren't if-else statements simply used to call methods that can throw exceptions? What do try-catch blocks offer that if-else statements do not?

Answer :

13. Why must you order catch blocks from smallest to largest? Specifically address the problem that you would encounter if you ordered exceptions from largest to smallest. Use the inheritance tree on page 332 to give an example of possible problem. (note: the compiler will complain even if you attempt to do this; this is just an exercise to see WHY the compiler complains).

Answer :

14. Refer to the Exception class hierarchy on page 332. Which of the following catch blocks is in the INCORRECT location?

```
try
{
    laundry.doLaundry();
}

catch(UniformException uniformEx)
{
}

catch(TeeShirtException tshirtEx)
{
}

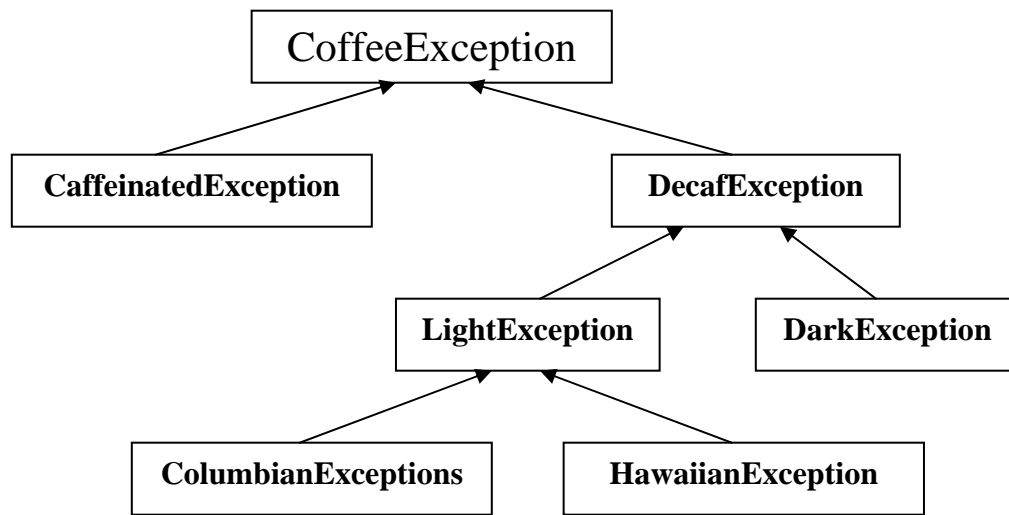
catch(ShirtException shirtEx)
{
}

catch(ClothingException clothingEx)
{
}

catch(LingerieException lingerieEx)
{
}
```

Answer :

15. Consider the following Exception class hierarchy.



Given the following try block and the CoffeeException hierarchy above,

```
try
{
    CoffeeDrinker.drinkCoffee(); //throws CoffeeException
}
```

Write all of the catch blocks needed to catch each of the exceptions in the hierarchy. Be sure to follow the “smallest to largest” rule for exceptions.

Answer :