Question 3:

When catching multiple exceptions in a single catch block separated by pipe, the exception types should be disjoint. In this case the FileNotFoundException is a subclass of IOException and hence cannot be placed together in a single catch block and must be handled separately as below.

```
import java.io.FileNotFoundException;
import java.io.IOException;

public class TestException {
   public static void main(String[] args) {
      try {
         testExceptions();
      } catch (FileNotFoundException e) {
         e.printStackTrace();
      } catch (IOException e) {
         e.printStackTrace();
      }
   }
   public static void testExceptions() throws IOException, FileNotFoundException{
   }
}
```

Question 4:

The NumberFormatException is a subclass of Exception class. In the order of catch blocks the subclasses must come first and then the superclass so that the most specific exception will be caught first before the general one. So the NumberFormatException catch block should be placed above the Exception catch block.

```
class ExceptionHandling
{
   public static void main (String[] arts)
   {
      try
      {
         int i = Integer.parseInt("abc");
      }
      catch(NumberFormatException ex)
      {
         System.out.println("");
      }
      catch(Exception ex)
      {
         System.out.println("This block handles all exception types");
      }
   }
}
```

Question 5:

throw: the throw keyword is used to explicitly throw an exeption. It is generally followed by an instance of Throwable object. After the throw statement is executed the program is stopped and it checks for the try catch statement that matches the exception.

throws: the throws keyword is used in method signature to indicate that the method might throw one of the exceptions listed in the method signature. This keyword is followed by a list of exceptions that the method might throw separated by commas. When sucha method is called the exceptions can be handled using try catch block.

Throwable: the throwable class is the super class of all errors and exceptions in java. Only those objects that are insances of this throwable class or its subclasses can be thrown by the java thrown statement or be the argument type in catch clause.