**Practical 2**

1. Create the following AndroidManifest.xml in Android Studio. Understand the xml file and extract the information to generate the similar application. You may create the necessary class in the exercise.



1. Create the following AndroidManifest.xml in Android Studio. Understand the xml file and extract the information to generate the similar application. You may create the necessary class in the exercise.



**Extra Information**

Information: (**http://developer.android.com/guide/practices/screens\_support.html)**

*Density-independent pixel (dp)*

A virtual pixel unit that you should use when defining UI layout, to express layout dimensions or position in a density-independent way.

The density-independent pixel is equivalent to one physical pixel on a 160 dpi screen, which is the baseline density assumed by the system for a "medium" density screen. At runtime, the system transparently handles any scaling of the dp units, as necessary, based on the actual density of the screen in use. The conversion of dp units to screen pixels is simple: px = dp \* (dpi / 160). For example, on a 240 dpi screen, 1 dp equals 1.5 physical pixels. You should always use dp units when defining your application's UI, to ensure proper display of your UI on screens with different densities.

**Extra Information**

Hardware features

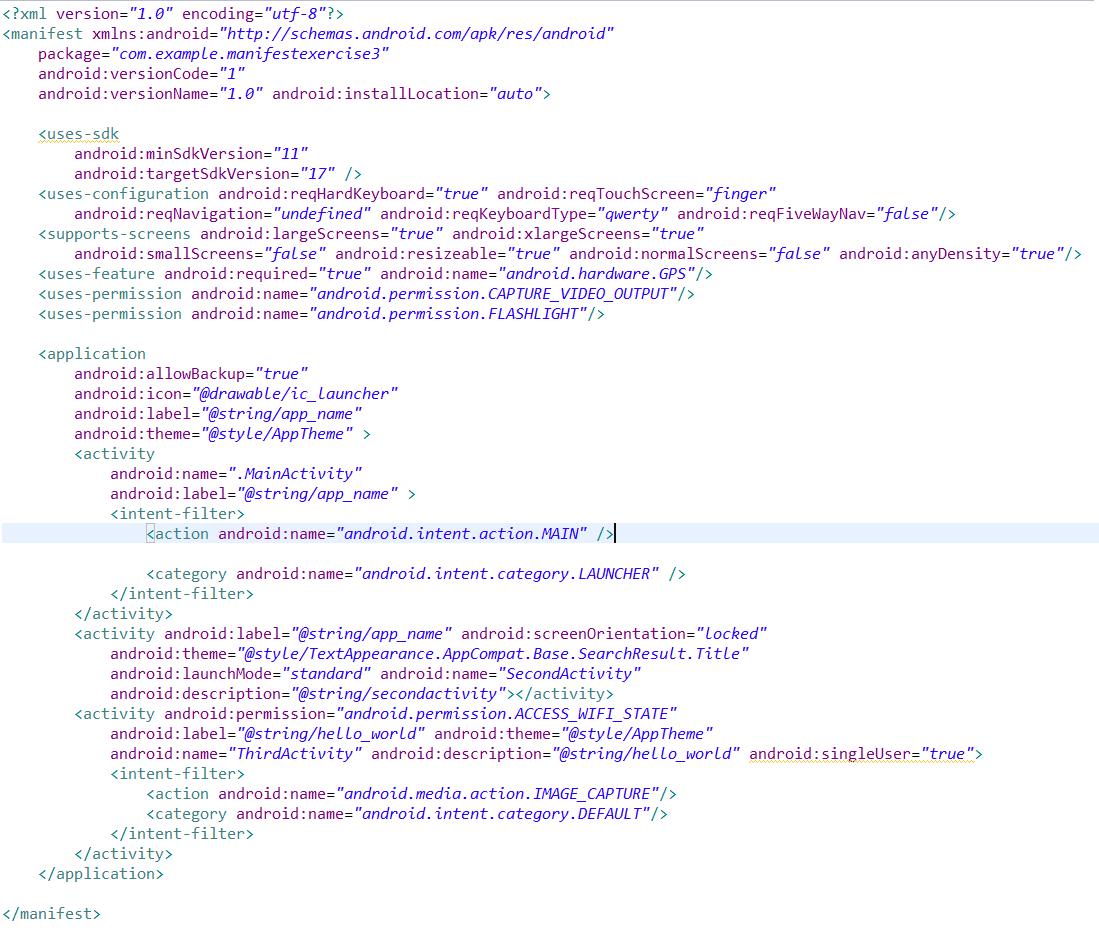
The table below describes the hardware feature descriptors supported by the most current platform release. To signal that your application uses or requires a hardware feature, declare each value in a android:name attribute in a separate <uses-feature> element.

**Source : http://developer.android.com/guide/topics/manifest/uses-feature-element.html**

1. Create the following AndroidManifest.xml in Android Studio. Understand the xml file and extract the information to generate the similar application. You may create the necessary class in the exercise.



1. Create the following AndroidManifest.xml in Android Studio. Understand the xml file and extract the information to generate the similar application. You may create the necessary class in the exercise.



1. Create the following AndroidManifest.xml in Android Studio. Understand the xml file and extract the information to generate the similar application. You may create the necessary class in the exercise.



**Question 6**

**import** java.util.Scanner;

**public** **class** AreaCalculationTester

{

**public** **static** **void** main(String []args)

{

String input;

**double** length;

**double** width;

Scanner in = **new** Scanner(System.*in*);

System.*out*.println("Enter the length ");

input=in.next();

length=Integer.*parseInt*(input);

System.*out*.println("Enter the width ");

input=in.next();

width=Integer.*parseInt*(input);

AreaCalculation obj1=**new** AreaCalculation(length,width);

System.*out*.println(obj1.getArea());

}

}

**public** **class** AreaCalculation

{

**private** **double** length;

**private** **double** width;

**public** AreaCalculation(**double** length, **double** width)

{

**this**.length=length;

**this**.width=width;

}

**public** **double** getArea()

{

**return** length\*width;

}

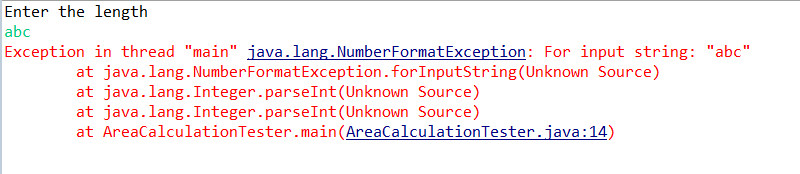
}

The coding provided above is a simple program to perform the calculation of area. However, this program do not handle exception such as **NumberFormatException** and also allows user to input a **NEGATIVE** input. The sample output 1 show the error when input is a letter. The sample output 2 shows the result of NEGATIVE value when user entered a negative value. You are required to modify the program so that it able to perform the following tasks:

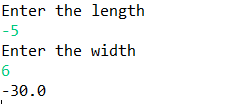
* You are required to modify the program so that it able to handle the **NumberFormatException** and **lessthanZERO** by showing the appropriate error message.
* You are required to create a **lessthanZERO** class so that when the input for length or width is less than 0, error message will be prompted.
* The program should continue prompt the user for the correct input if the entered input is not correct.

The sample output 3 shows the execution of the program after the modification

**Sample Output 1**



**Sample Output 2**



**Sample Output 3**

