Question 1:

a) What should you do disable the effect of the back button?
Overriding the onBackPressed() method in the Activity Instance with nothing or customised actions and not calling the super.onBackPressed() method. For example:

```
@Override
public void onBackPressed() {
    Toast.makeText(this, "This is a back button", Toast.LENGTH_SHORT).show();
}
```

b) What should you do to force your Android app not to save views data during device reorientation?
Overriding the onSaveInstanceState() method with an if statement that will never be true and put the super.onSaveInstanceState() in there.

```
@Override
protected void onSaveInstanceState(@NonNull Bundle outState) {
    if (false)
        super.onSaveInstanceState(outState);
    else {
        //do nothing
    }
}
```

Question 2:

a) What is the effect of calling super() in Lifecycle callbacks? support your answer with examples
The supers perform a lot of standard processing required by any Activity. Without calling the super, the required base for our customised code will not be run, therefore the whole activity will not function properly. For example:

```
@Override
protected void onRestoreInstanceState(@NonNull Bundle savedInstanceState) {
    //super.onRestoreInstanceState(savedInstanceState);
    Log.i("MyLifeCycleMethods","This is onRestoreInstance");
}
```

Without the super.onRestoreInstanceState(), the saved view state of the activity will not be stored on the Activity but the log is still recorded.

b) What is the difference between getPreferences() and getSharedPreferences()?
getPreferences is used from an Activity when you only need to use only one shared preference file for the activity. As it retrieves a default shared preference file that belongs to the activity, supplied name is not required.
getSharedPreferences is used if multiple shared preference files identified by names. This method can be called from any Context in the app.

c) Write a piece of code that stores the following key-value pairs in a shared preferences file named “fit2081w3.data”
    a. Age: 23
    b. Name: “Lara”

```java
public class MainActivity extends AppCompatActivity {
    EditText etName;
    EditText etAge;
    Button btnShow;
    int timesClicked;
    SharedPreferences userInfo;

    public void saveInfo(){
        SharedPreferences.Editor edit = userInfo.edit();
        edit.putString("name","Lare");
        edit.putInt("age", 23);
        edit.apply();
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        userInfo = getSharedPreferences("fit2081w3.data",MODE_PRIVATE);

        etName = findViewById(R.id.etName);
        btnShow = findViewById(R.id.btnShow);
        etAge = findViewById(R.id.etAge);


        btnShow.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                saveInfo();
            }
        });
    }

}
```

Question 3:

a) What is the difference between task and activity in Android applications?
Activity is a single screen with user-interface that user can interact with. Whereas a task is a collection of activities that users interact with when performing a certain job.

b) Briefly explain one similarity and one difference between shared preferences and bundle in Android applications.
Similarity: They all save/share data between activities.
Difference: Data saved by shared preferences is still there even when the app is closed. Whereas for bundle, as soon as the app is closed, its data will be erased.