PRACTICAL-3

AIM- Develop an Android Application to demonstrate the Activity Life Cycle.

The Activity base class defines a series of events that govern the life cycle of an activity. The Activity class defines the following events:

onCreate() — Called when the activity is first created

onStart() — Called when the activity becomes visible to the user

onResume() — Called when the activity starts interacting with

the user

onPause() — Called when the current activity is being paused and the previous activity is being

resumed onStop() — Called when the activity is no longer visible to the user

onDestroy() — Called before the activity is destroyed by the system

onRestart() — Called when the activity has been stopped and is restarting

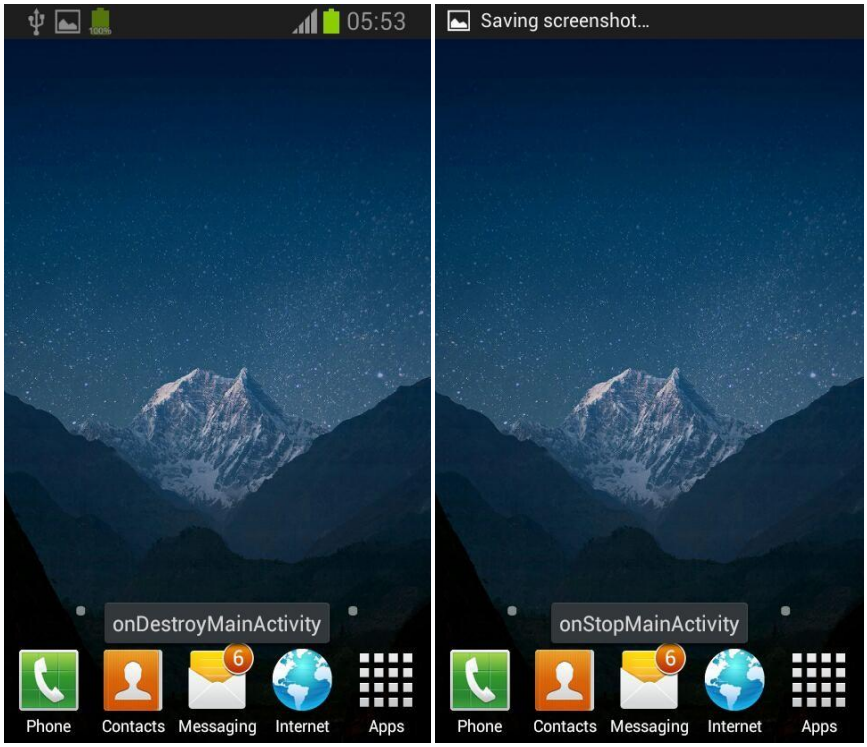
again . By default, the activity created for you contains the onCreate()

event.

```
packagecom.example.activity_life;

importandroid.annotation.SuppressLint;
importandroid.app.Activity; importandroid.os.Bundle;
importandroid.widget.Toast;

@SuppressLint("NewApi")
publicclassMainActivityextends Activity {

@Override
protectedvoidonCreate(Bundle savedInstanceState){
super.onCreate(savedInstanceState); notify("onCreate");
}

@Override
protectedvoidonPause(){
super.onPause();
notify("onPause");
}

@Override
protectedvoidonResume(){
```

```
super.onResume();
notify("onResume");
}

@Override
protectedvoidonStop(){
super.onStop();
notify("onStop");


}

    @Override
protectedvoidonDestroy(){
super.onDestroy();
        notify("onDestroy");
}

    @Override
protectedvoidonRestoreInstanceState(Bundle savedInstanceState){
super.onRestoreInstanceState(savedInstanceState); notify("onRestoreInstanceState");
}

    @Override
protectedvoidonSaveInstanceState(Bundle outState){
super.onSaveInstanceState(outState); notify("onSaveInstanceState");
}

privatevoidnotify(String methodName){
String name =this.getClass().getName();
String[] strings = name.split("\\.");
        Toast.makeText(getApplicationContext(),
                methodName+""+ strings[strings.length-1],
                Toast.LENGTH_LONG).show();
}

}
```

OUTPUT:-


05:53
onDestroyMainActivity
Phone
Contacts
Messaging
Internet
Apps
Saving screenshot...
onStopMainActivity
Phone
Contacts
Messaging
Internet
Apps

PRACTICAL-4

AIM- Write a C program to implement CRC.

Code:-

```
#include <stdio.h>

#include<conio.h>

#include<string.h>

 void main() {

        int i,j,keylen,msglen;

        char input[100], key[30],temp[30],quot[100],rem[30],key1[30];

        clrscr();

        printf("Enter Data: ");

        gets(input);

        printf("Enter Key: ");

        gets(key);

        keylen=strlen(key);

        msglen=strlen(input);

        strcpy(key1,key);

        for (i=0;i<keylen-1;i++) {

                input[msglen+i]='0';

        }

        for (i=0;i<keylen;i++)
```

```
 temp[i]=input[i];
for (i=0;i<msglen;i++) {
        quot[i]=temp[0];
        if(quot[i]=='0')
for (j=0;j<keylen;j++)
 key[j]='0'; else
         for (j=0;j<keylen;j++)
         key[j]=key1[j];
        for (j=keylen-1;j>0;j--) {
                if(temp[j]==key[j])
                 rem[j-1]='0'; else
                 rem[j-1]='1';
        }
        rem[keylen-1]=input[i+keylen];
        strcpy(temp,rem);
}
strcpy(rem,temp);
printf("\nQuotient is ");
for (i=0;i<msglen;i++)
 printf("%c",quot[i]);
printf("\nRemainder is ");
for (i=0;i<keylen-1;i++)
 printf("%c",rem[i]);
```

```
        printf("\nFinal data is: ");

        for (i=0;i<msglen;i++)

         printf("%c",input[i]);

        for (i=0;i<keylen-1;i++)


         printf("%c",rem[i]);

        getch();

    }
```

Output: