T.C.
PİRİ REİS UNIVERSITY
MARITIME HIGHER VOCATIONAL SCHOOL
DEPARTMENT OF COMPUTER TECHNOLOGIES
COMPUTER PROGRAMMING PROGRAMME
FALL SEMESTER OF 2019-2020 ACADEMIC YEAR
BIP2019 MOBILE PROGRAMMING FINAL EXAM

(Following items will be completed in ink) Date: January 08, 2020

Student's Name&Surname : ....

PRU student number : ....
Class : ....

Signature : ....

A

INSTRUCTIONS:

1. No student shall be permitted to enter the examination room after the first 20 minutes.
2. No student shall be permitted to leave during the first 20 minutes of the examination.
3. Students are not permitted to ask questions to the invigilators, except in cases of supposed errors or ambiguities in examination questions.
4. Students **suspected** of any of the following, or similar, **dishonest practices** will be **immediately dismissed from the examination** and will be subject to disciplinary action:
   - Having written and/or printed materials other than authorized by the examiners.
   - Speaking or communicating with other students.
5. Mobile phones and all electronic devices (e.g. electronic watches etc.) won't be in the exam room. It can be taken off on the table if the exam supervisor allow in surveillance control.
6. All answers must be written down on the designated place **or** the blank place following the questions on the examination paper. (Additional blank papers may be handed in by instructor when it is required)

EACH STUDENT SHOULD BE PREPARED TO INTRODUCE, UPON REQUEST, HIS/HER STUDENT/ID CARD. ALL YOUR WRITING MUST BE HANDED IN. LECTURES SHOULD READ EXAM İNSTRUCTION BEFORE START OF THE EXAM.

Number of questions : 1 Time allocated : Take home exam Assessment Weighting : % 45 Grading criteria : The weight of each question is written next to it. Instructor : Assist. Prof. Dr. Pınar Demir Instructor's signature :	Grade :............................ *Remarks about student's performance:*

GOOD LUCK!!

QUESTIONS:

See attached sheet for questions.

Q- Create an Android Stop Watch application, which has basic features of a stop watch like, Start, Stop, Pause and Reset.

Java Files

MainActivity.java

```
package com.mertkaraca.chronomerter;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.os.SystemClock;
import android.view.View;
import android.view.WindowManager;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.Chronometer;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {

    /*Creating our variables globally to access
    them everywhere we want, We'll to determine them later*/
    Button btnStart,btnPause,btnResume,btnReset;
    Chronometer chronometer;
    Animation rotate;
    ImageView imageView;
    long pause;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //Keeps screen on while view is open
        getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);

        //Determining variables values with their own id's which we dedicated in
activity_main.xml
        btnStart = findViewById(R.id.btnStart);
        btnPause = findViewById(R.id.btnPause);
        btnResume = findViewById(R.id.btnResume);
        btnReset = findViewById(R.id.btnReset);
        chronometer = findViewById(R.id.chronometer);
        imageView = findViewById(R.id.imageView);

        //This object getting it's functionality from rotation.xml
        rotate = AnimationUtils.loadAnimation(MainActivity.this,R.anim.rotation);

        //Start Button
        btnStart.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //our imageView start to animate with rotate which we wrote part of
it in rotation.xml
                imageView.startAnimation(rotate);
                chronometer.setFormat("%s");
                //chronometer object subtract pause's value from elapsed time
                chronometer.setBase(SystemClock.elapsedRealtime()-pause);
                chronometer.start();
                //After we start chronometer object start button going back and 
finish button come forward
                btnStart.setVisibility(View.INVISIBLE);
                btnPause.setVisibility(View.VISIBLE);
            }
        });

        //Stop Button
        btnPause.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //Stops chronometer visually
                chronometer.stop();
                //We are using this expression to use it Start and Resume buttons
                pause = SystemClock.elapsedRealtime() - chronometer.getBase();
                //Stops and resets the animation
                imageView.clearAnimation();

                //When we press the Stop button, the Stop button is invisible,
                //and the Resume and Finish buttons are visible.
                btnPause.setVisibility(View.INVISIBLE);
                btnResume.setVisibility(View.VISIBLE);
                btnReset.setVisibility(View.VISIBLE);
            }
        });

        //Resume Button
        btnResume.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //We should use this expression to resume the chronometer
                //It basically prevent chronometer from elapsed time
                chronometer.setBase(SystemClock.elapsedRealtime()-pause);
                //Activates chronometer visually
                chronometer.start();
                //Starts turning animation
                imageView.startAnimation(rotate);

                //When we press the Resume button, the Resume and Reset buttons 
become invisible,
                //and the Pause button becomes visible.
                btnResume.setVisibility(View.INVISIBLE);
                btnReset.setVisibility(View.INVISIBLE);
                btnPause.setVisibility(View.VISIBLE);
            }
        });

        //Reset Button
        btnReset.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                chronometer.stop();
                //It set's chronometer's base value to default to restart the process
                chronometer.setBase(SystemClock.elapsedRealtime());
                //We equaling pause to default value to restart the process again
                pause = 0;
                //imageView stop's rotating
                rotate.cancel();
                imageView.setAnimation(rotate);

                //When we press the Reset button, the Resume and Finish buttons 
become invisible
```

```
                //and the Start button becomes visible.
                btnReset.setVisibility(View.INVISIBLE);
                btnResume.setVisibility(View.INVISIBLE);
                btnStart.setVisibility(View.VISIBLE);
            }
        });
    }
}
```

Layout Files

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Constraint Layout (All default except background color -->
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/black"
    tools:context=".MainActivity">

    <!-- Chronometer -->
    <Chronometer
        android:id="@+id/chronometer"
        android:layout_width="300dp"
        android:layout_height="100dp"
        android:fontFamily="sans-serif"
        android:gravity="center"
        android:text="00:00"
        android:textColor="@color/white"
        android:textSize="60dp"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@id/imageView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <!-- Image View -->
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:src="@drawable/circle"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
```

```xml
<!-- Start Button -->
   <Button
       android:id="@+id/btnStart"
       android:layout_width="250dp"
       android:layout_height="250dp"
       android:text="START"
       android:textColor="@color/white"
       android:textStyle="bold"
       android:textSize="30dp"
       android:fontFamily="monospace"
       android:visibility="visible"
       android:stateListAnimator="@null"
       android:background="@drawable/circle_button"
       app:layout_constraintBottom_toBottomOf="@+id/imageView"
       app:layout_constraintEnd_toEndOf="@+id/imageView"
       app:layout_constraintStart_toStartOf="@+id/imageView"
       app:layout_constraintTop_toTopOf="@+id/imageView" />

   <!-- Pause Button -->
   <Button
       android:id="@+id/btnPause"
       android:layout_width="250dp"
       android:layout_height="250dp"
       android:text="PAUSE"
       android:textColor="@color/white"
       android:textStyle="bold"
       android:textSize="30dp"
       android:fontFamily="monospace"
       android:visibility="invisible"
       android:stateListAnimator="@null"
       android:background="@drawable/circle_button_yellow"
       app:layout_constraintBottom_toBottomOf="@+id/imageView"
       app:layout_constraintEnd_toEndOf="@+id/imageView"
       app:layout_constraintStart_toStartOf="@+id/imageView"
       app:layout_constraintTop_toTopOf="@+id/imageView" />

   <!-- Resume Button -->
   <Button
       android:id="@+id/btnResume"
       android:layout_width="125dp"
       android:layout_height="250dp"
       android:background="@drawable/circle_button_green"
       android:fontFamily="monospace"
       android:text="RESUME"
       android:textColor="@color/white"
       android:textSize="30dp"
       android:textStyle="bold"
       android:visibility="invisible"
       app:layout_constraintBottom_toBottomOf="@+id/imageView"
       app:layout_constraintEnd_toEndOf="@+id/imageView"
       app:layout_constraintHorizontal_bias="0.142"
       app:layout_constraintStart_toStartOf="@+id/imageView"
       app:layout_constraintTop_toTopOf="@+id/imageView"
       app:layout_constraintVertical_bias="0.497" />
```

```
 <!-- Reset Button -->
    <Button
        android:id="@+id/btnReset"
        android:layout_width="125dp"
        android:layout_height="250dp"
        android:background="@drawable/circle_button_red"
        android:fontFamily="monospace"
        android:text="RESET"
        android:textColor="@color/white"
        android:textSize="30dp"
        android:textStyle="bold"
        android:visibility="invisible"
        app:layout_constraintBottom_toBottomOf="@+id/imageView"
        app:layout_constraintEnd_toEndOf="@+id/imageView"
        app:layout_constraintHorizontal_bias="0.851"
        app:layout_constraintStart_toStartOf="@+id/imageView"
        app:layout_constraintTop_toTopOf="@+id/imageView"
        app:layout_constraintVertical_bias="0.497" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Animation Files

Linear_interpolator.xml

```
<?xml version="1.0" encoding="utf-8"?>
<linearInterpolator xmlns:android="http://schemas.android.com/apk/res/android"/>
<!-- We created this linear interpolator to use it in rotation.xml -->
```

Rotation.xml

```
<?xml version="1.0" encoding="utf-8"?>
<rotate xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromDegrees="0"
    android:toDegrees="360"
    android:pivotX="50%"
    android:pivotY="50%"
    android:duration="10000"
    android:interpolator="@anim/linear_interpolator"
    android:repeatCount="infinite">
    <!-- We gave duration as 10000 but it's not accurate. All this animation just for
fun actually-->
    <!-- The magic behind the animating black circle is it just turning 360 degree in
dedicated duration. Not a big deal as you see -->
</rotate>
```

Drawable Files

Circle_button.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="@color/green"/>
    <corners android:radius="900dp"/>
    <!-- We created this shape to use it inside the yellow circle -->
</shape>
```

Cirlce_button_yellow.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="@color/yellow"/>
    <corners android:radius="900dp"/>
</shape>
```

Circle_button_green.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <!-- Determining the shapes general width and height-->
    <size android:height="1000dp"
        android:width="500dp"/>
    <!-- Determining the color-->
    <solid android:color="@color/green"/>
    <!-- Giving radius to catch half round -->
    <corners android:topLeftRadius="500dp"
        android:bottomLeftRadius="500dp"/>
</shape>
```

Circle_button_red.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
android:shape="rectangle">
    <!-- Determining the shapes general width and height-->
    <size android:height="1000dp"
    android:width="500dp"/>
    <!-- Determining the color-->
    <solid android:color="@color/red"/>
    <!-- Giving radius to catch half round -->
    <corners android:topRightRadius="500dp"
    android:bottomRightRadius="500dp"/>
</shape>
```

Values Files

Colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <!--We change this colors to make better visual our view-->
    <color name="colorPrimary">#000</color>
    <color name="colorPrimaryDark">#000</color>
    <color name="colorAccent">#FFC72B</color>

    <!--We define this colors to use them later-->
    <color name="yellow">#FFC72B</color>
    <color name="darkGrey">#1F1F1F</color>
    <color name="white">#fff</color>
    <color name="black">#000</color>
    <color name="red">#DC3545</color>
    <color name="green">#28A745</color>
</resources>
```

Styles.xml

```
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>
</resources>
```