UNIVERSITI TUNKU ABDUL RAHMAN
Faculty of Information and Communication Technology

UCCD3223 Mobile Applications Development
(Jan 2024 Trimester)

Individual Practical Assignment

Name	**Chu Wai Hao**
Student ID	**21ACB02803**
Course	**IB**
Practical Group	**P1**
Lecturer	**Mr Tan Chiang Kang**

Marking scheme	Marks	Remarks
Correctness	× 2.5	
Design	× 3.5	
User Friendliness	× 2	
Neat Program Documentation		
Report Format		
TOTAL		

Introduction

This mobile application is designed for primary school students and focuses on mathematic topics by using java language. The number used are limited to 3 digits and whole numbers only. The mobile application covers three main topics which is comparing numbers, ordering numbers, and composing numbers. This mobile application will present questions in quiz format for user to answer. Then this mobile application consists of 4 pages which is the welcome page, exercise selection page, the question display page, and the result page.

Figure1.1

Figure 1.1 to show the welcome page. When the users open this mathematic mobile application, this is the first thing they see. To begin the quiz, users can click on the " Start Quiz" button when they're ready.

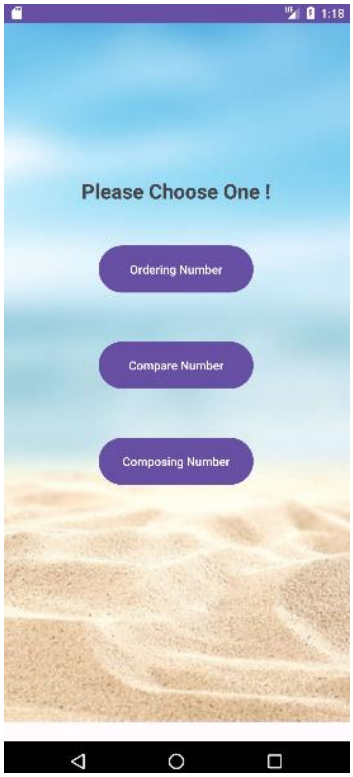


Figure 1.2

Figure 1.2 to show the exercise selection page. When users click the “Start Quiz” button, they will see this page. Here, users can choose the type of exercise they want to do. They can select from three options: “Ordering Numbers”, “Comparing Numbers”, and “Composing Numbers”.

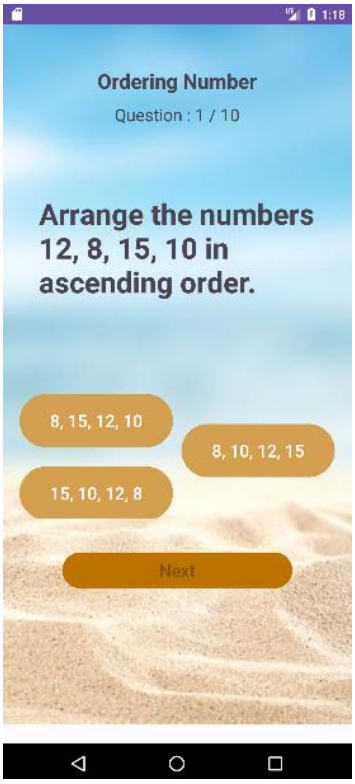


Figure 1.3

Figure 1.3 to show the question page. After users choose an exercise from the exercise selection screen, questions similar to Figure 1.3 will appear. Each exercise comprises 10 questions, and the system shows them randomly to ensure users don't get the same question repeatedly.

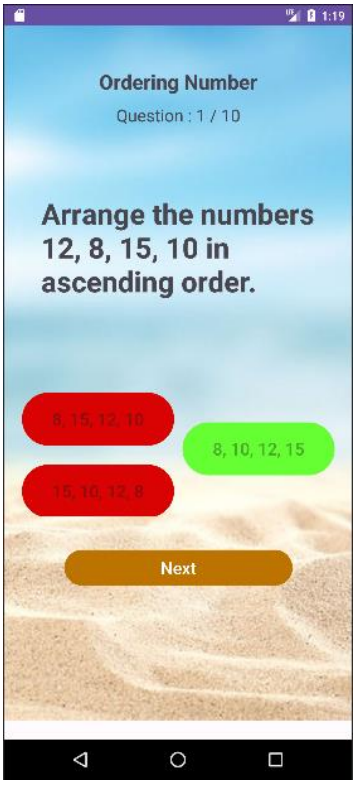


Figure 1.4

Figure 1.4 illustrates what happens after the user selects an answer. The system will announce which answer is correct, highlighting it in green, while incorrect answers will be shown in red.

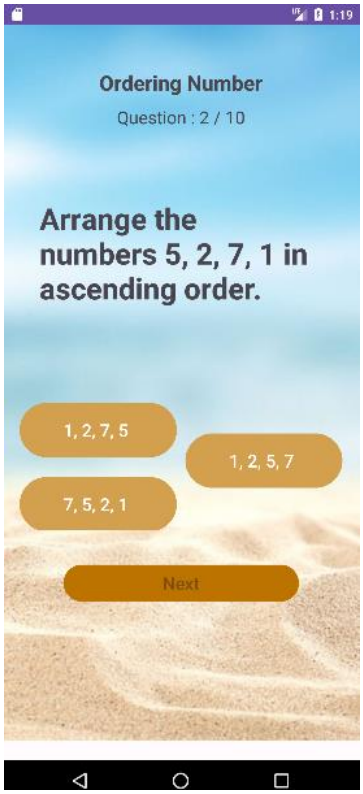
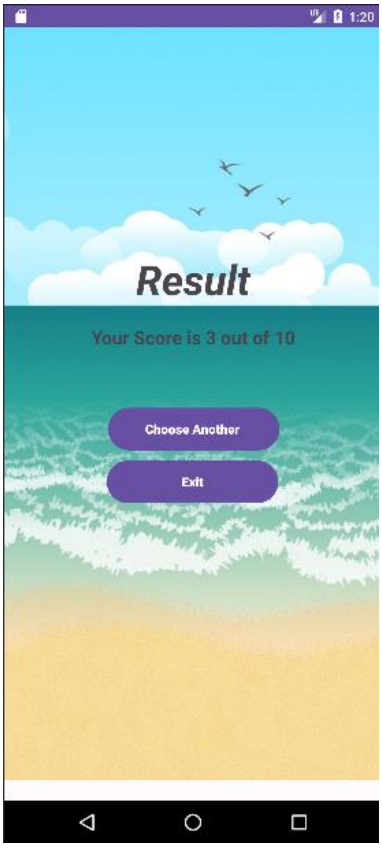


Figure 1.5

Once the user has completed answering question 1 and the system has shown the correct answer, they can click the “Next” button to move to question 2. Figure 1.5 displays question 2, allowing the user to proceed to the next question.



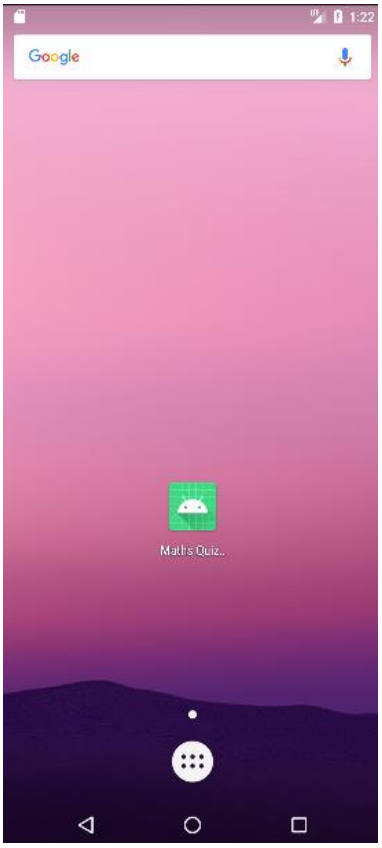


Figure 1.6

Figure 1.7

Figure 1.6 is showing the result page at the end of the questions. Here, the system shows the total number of correct answers. In simple terms, this page presents the user's result. There are two buttons for the user to proceed further. The user can choose "Choose Another" to go back to the exercise page and select a different exercise. Alternatively, selecting "Exit" means the user will leave the mobile application, as shown in Figure 1.7.

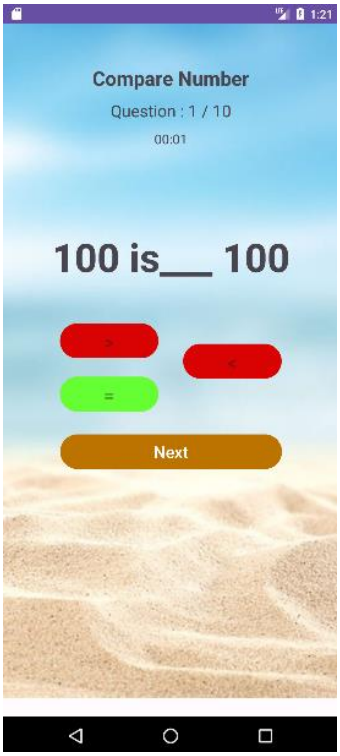


Figure 1.8

If the user clicks “Choose Another” in Figure 1.6, the system will show Figure 1.2 again for the user to select another exercise. For instance, if the user chooses the “Compare Number” exercise, Figure 1.8 will display the first question related to comparing numbers. When the user selects the correct number, the system will show it in green. However, if the answer is wrong, it displays in red.

Figure 1.9

Figure 1.9 presents a question form the third exercise, which focuses on “Composing Number”.

Conclusion

In the conclusion, this mobile application is designed for primary school students to improve their math skills in a fun and interactive way. It offers three main exercise: comparing numbers, ordering numbers and composing numbers. The app uses a quiz format with randomly generated questions to keep the learning experience engaging. With features such as color-coded feedback for correct and incorrect answers, as well as options to select different exercises and track progress, this app aims to make learning math enjoyable and effective for young learners.

Source Code:

Main Avtivity

```
public class MainActivity extends AppCompatActivity {
    Button start;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      start = findViewById(R.id.button);
      start.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            Intent homepage = new Intent(MainActivity.this,ChooseTopic.class);
            startActivity(homepage);

         }
      });


   }
```

Choose Topic

```
   public class ChooseTopic extends AppCompatActivity {
      Button Zero,One,Two;

      @Override
      protected void onCreate(Bundle savedInstanceState) {
         super.onCreate(savedInstanceState);
         setContentView(R.layout.activity_choose_topic);

         Zero = findViewById(R.id.zero);
         One = findViewById(R.id.one);
         Two = findViewById(R.id.two);

         Zero.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
               Intent on = new Intent(ChooseTopic.this,OrderNumActivity.class);
               startActivity(on);
            }
         });
         One.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
            public void onClick(View v) {
                Intent  cn = new Intent(ChooseTopic.this,ComNumActivity.class);
                startActivity(cn);
            }
        });
        Two.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent cp = new Intent(ChooseTopic.this,ComposingNumActivity.class);
                startActivity(cp);
            }
        });
    }
}
```

Compare Number Activity

```
public class ComNumActivity extends AppCompatActivity implements
View.OnClickListener{

    TextView level,questionNumber, timer, question;
    Button opt1, opt2, opt3, submit;

    int questionIndex = 0;
    private int score = 0;

    int totalQuestion = ComNumQues.question1.length;
    String selectedAns = "";


    CountDownTimer countDownTimer;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_compare_number);


        //Connect using an XML tag that has an ID attribute.
        level = findViewById(R.id.Two);
        questionNumber = findViewById(R.id.ttl_questions);
        timer = findViewById(R.id.timelimit);
        question = findViewById(R.id.question1);

        opt1 = findViewById(R.id.option11);
        opt2 = findViewById(R.id.option21);
```

```
    opt3 = findViewById(R.id.option31);
    submit = findViewById(R.id.submit1);

    opt1.setOnClickListener(this);
    opt2.setOnClickListener(this);
    opt3.setOnClickListener(this);
    submit.setOnClickListener(this);
    questionNumber.setText("Total questions: " + totalQuestion);
    submit.setEnabled(false);//Initially, the submission feature should be disabled.


    //set level text
    level.setText("Compare Number");

    //load question
    loadNewQuestion();
}

//getRandomNonRepeatingIntegers function
public static ArrayList getRandomNonRepeatingIntegers(int size, int min, int max) {
    ArrayList numbers = new ArrayList();
    Random random = new Random();
    while (numbers.size() < size) {
        int randomNumber = random.nextInt((max - min) + 1) + min;
        if (!numbers.contains(randomNumber)) {
            numbers.add(randomNumber);
        }
    }
    return numbers;
}

//on click function
@Override
public void onClick(View v) {


    Button clickButton = (Button) v;

    if (clickButton.equals(submit)){

        if (selectedAns.equals(ComNumQues.answer1[(int) List.get(questionIndex)])){
            score++;
        }

        questionIndex++;
        loadNewQuestion();
```

```
            }
            else {
                countDownTimer.cancel();
                selectedAns = clickButton.getText().toString();
                if (selectedAns == null) {
                    submit.setEnabled(false); //If the selectedAnswer is null, then the submit
button should be disabled
                } else {
                    submit.setEnabled(true); //If the the selectedAnswer is not null, then the
submit button be enable
                }
                showAns();

            }
        }

        //ArrayList List = getRandomNonRepeatingIntegers(questions.questions.length, 0,
(questions.questions.length-1));
        ArrayList List = getRandomNonRepeatingIntegers(ComNumQues.question1.length,
0, (ComNumQues.question1.length-1));
        void loadNewQuestion(){
            timer();
            if (questionIndex == totalQuestion) {
                finishQuiz();
                return;
            }
            submit.setEnabled(false);
            opt1.setEnabled(true);
            opt2.setEnabled(true);
            opt3.setEnabled(true);



            opt1.setBackgroundColor(Color.parseColor("#d3a04f"));
            opt2.setBackgroundColor(Color.parseColor("#d3a04f"));
            opt3.setBackgroundColor(Color.parseColor("#d3a04f"));



            questionNumber.setText("Question : "+(questionIndex+1)+" / "+totalQuestion );
            question.setText(ComNumQues.question1[(int) List.get(questionIndex)]);
            opt1.setText(ComNumQues.choice1[(int) List.get(questionIndex)][0]);
            opt2.setText(ComNumQues.choice1[(int) List.get(questionIndex)][1]);
            opt3.setText(ComNumQues.choice1[(int) List.get(questionIndex)][2]);

        }
```

```
//finish quiz function
void finishQuiz(){
   countDownTimer.cancel();
   Intent intent = new Intent(ComNumActivity.this,ResultScore.class);
   intent.putExtra("score",score);
   startActivity(intent);
   finish();
}

void showAns(){
   String b1=opt1.getText().toString();
   String b2=opt2.getText().toString();
   String b3=opt3.getText().toString();


   opt1.setEnabled(false);
   opt2.setEnabled(false);
   opt3.setEnabled(false);



   //opt1
   if (b1.equals(ComNumQues.answer1[(int) List.get(questionIndex)])){
      opt1.setBackgroundColor(Color.parseColor("#66ff33"));
   }
   else {opt1.setBackgroundColor(Color.parseColor("#d80303"));}

   //opt2
   //if (b2.equals(questions.answer[(int) List.get(questionIndex)])){
   if (b2.equals(ComNumQues.answer1[(int) List.get(questionIndex)])){
      opt2.setBackgroundColor(Color.parseColor("#66ff33"));
   }
   else {opt2.setBackgroundColor(Color.parseColor("#d80303"));}

   //opt3
   //if (b3.equals(questions.answer[(int) List.get(questionIndex)])){
   if (b3.equals(ComNumQues.answer1[(int) List.get(questionIndex)])){
      opt3.setBackgroundColor(Color.parseColor("#66ff33"));
   }
   else {opt3.setBackgroundColor(Color.parseColor("#d80303"));}



}
private void timer() {
```

```
        countDownTimer = new CountDownTimer(20000, 1000) {
            @Override
            public void onTick(long millisUntilFinished) {
                // calculate seconds remaining and format as a string
                String seconds = String.format("%02d", millisUntilFinished / 1000);
                timer.setText("00:" + seconds);
            }

            @Override
            public void onFinish() {
                questionIndex++;
                loadNewQuestion();
            }
        }.start();
    }
}
```

Compare Number Questions

```
public class ComNumQues {

    public static String question1[]={
            "934 is ___ 234",
            "284 is___ 697",
            "651 is___ 567",
            "42 is___ 66",
            "100 is___ 100",
            "378 is___ 293",
            "127 is___ 148",
            "234 is___ 243",
            "56 is___ 567",
            "198 is___ 93"
    };

    public static String choice1[][]={
            {">","<","="},
            {">","<","="},
            {">","<","="},
            {">","<","="},
            {">","<","="},
            {">","<","="},
            {">","<","="},
            {">","<","="},
            {">","<","="},
            {">","<","="},
    };
```

```
    public static String answer1[]={
          ">","<",">","<","=",">","<","<","<",">"
    };
}
```

Composing Number Activity

```
public class ComposingNumActivity extends AppCompatActivity implements
View.OnClickListener{
    TextView level,questionNumber, timer, question;
    Button opt1, opt2, opt3, submit;

    int questionIndex = 0;
    private int score = 0;
    int totalQuestion = ComposingNumQues.question3.length;

    String selectedAns = "";


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_composing_number);

        //Connect using an XML tag that has an ID attribute.
        level = findViewById(R.id.level2);
        questionNumber = findViewById(R.id.questionindex2);
        timer = findViewById(R.id.timer2);


        question = findViewById(R.id.question2);


        question = findViewById(R.id.question2);
        opt1 = findViewById(R.id.Option1);
        opt2 = findViewById(R.id.Option2);
        opt3 = findViewById(R.id.Option3);
        submit = findViewById(R.id.submit2);

        opt1.setOnClickListener(this);
        opt2.setOnClickListener(this);
        opt3.setOnClickListener(this);
        submit.setOnClickListener(this);
        questionNumber.setText("Total questions: " + totalQuestion);
        submit.setEnabled(false);//Initially, the submission feature should be disabled.
```

```
        //set level text
        level.setText("Composing Number");

        //load question
        loadNewQuestion();
    }
    public static ArrayList getRandomNonRepeatingIntegers(int size, int min, int max) {
        ArrayList numbers = new ArrayList();
        Random random = new Random();
        while (numbers.size() < size) {
            int randomNumber = random.nextInt((max - min) + 1) + min;
            if (!numbers.contains(randomNumber)) {
                numbers.add(randomNumber);
            }
        }
        return numbers;
    }
    @Override
    public void onClick(View v) {
        Button clickButton = (Button) v;

        if (clickButton.equals(submit)){
            if (selectedAns.equals(ComposingNumQues.answer3[(int)
List.get(questionIndex)])){
                score++;
            }

            questionIndex++;
            loadNewQuestion();
        }
        else {

            selectedAns = clickButton.getText().toString();
            if (selectedAns == null) {
                submit.setEnabled(false); //If the selectedAnswer is null, then the submit
button should be disabled
            } else {
                submit.setEnabled(true); //If the the selectedAnswer is not null, then the
submit button be enable
            }
            showAns();

        }
    }
```

```
    ArrayList List =
getRandomNonRepeatingIntegers(ComposingNumQues.question3.length, 0,
(ComposingNumQues.question3.length-1));
    void loadNewQuestion(){

      if (questionIndex == totalQuestion) {
        finishQuiz();
        return;
      }
      submit.setEnabled(false);
      opt1.setEnabled(true);
      opt2.setEnabled(true);
      opt3.setEnabled(true);

      opt1.setBackgroundColor(Color.parseColor("#d3a04f"));
      opt2.setBackgroundColor(Color.parseColor("#d3a04f"));
      opt3.setBackgroundColor(Color.parseColor("#d3a04f"));


      questionNumber.setText("Question : "+(questionIndex+1)+" / "+totalQuestion );
      question.setText(ComposingNumQues.question3[(int) List.get(questionIndex)]);
      opt1.setText(ComposingNumQues.choice3[(int) List.get(questionIndex)][0]);
      opt2.setText(ComposingNumQues.choice3[(int) List.get(questionIndex)][1]);
      opt3.setText(ComposingNumQues.choice3[(int) List.get(questionIndex)][2]);
    }

    //finish quiz function
    void finishQuiz(){

      Intent intent = new Intent(ComposingNumActivity.this,ResultScore.class);
      intent.putExtra("score",score);
      startActivity(intent);
      finish();
    }

    void showAns(){
      String b1=opt1.getText().toString();
      String b2=opt2.getText().toString();
      String b3=opt3.getText().toString();

      opt1.setEnabled(false);
      opt2.setEnabled(false);
      opt3.setEnabled(false);
```

```
        //opt1
        if (b1.equals(ComposingNumQues.answer3[(int) List.get(questionIndex)])){
            opt1.setBackgroundColor(Color.parseColor("#66ff33"));
        }
        else {opt1.setBackgroundColor(Color.parseColor("#d80303"));}

        //opt2
        if (b2.equals(ComposingNumQues.answer3[(int) List.get(questionIndex)])){
            opt2.setBackgroundColor(Color.parseColor("#66ff33"));
        }
        else {opt2.setBackgroundColor(Color.parseColor("#d80303"));}

        //opt3
        if (b3.equals(ComposingNumQues.answer3[(int) List.get(questionIndex)])){
            opt3.setBackgroundColor(Color.parseColor("#66ff33"));
        }
        else {opt3.setBackgroundColor(Color.parseColor("#d80303"));}

    }
}
```

Composing Number Questions

```
public class ComposingNumQues {
    public static String question3[]={
            "Compose the number 8 using two smaller numbers.",
            "Compose the number 12 using two smaller numbers.",
            "Compose the number 10 using two smaller numbers.",
            "Compose the number 15 using two smaller numbers.",
            "Compose the number 18 using two smaller numbers.",
            "Compose the number 20 using two smaller numbers.",
            "Compose the number 14 using two smaller numbers.",
            "Compose the number 16 using two smaller numbers.",
            "Compose the number 25 using two smaller numbers.",
            "Compose the number 30 using two smaller numbers."
    };

    public static String choice3[][]={
            {"4+2","5 + 3","3+1"},
            {"3 + 5","4 + 6","7 + 5"},
            {"6 + 4","2 + 3","9 + 3"},
            {"5 + 3","9 + 6","3 + 6"},
            {"2 + 8","10 + 8","13 + 5"},
            {"14 + 6","2 + 7","12 + 8"},
            {"6 + 8","2 + 4","12 + 11"},
            {"3 + 17","1 + 6","9 + 7"},
```

```
            {"8 + 9","15 + 10","10 + 10"},
            {"18 + 12","12 + 12","18 + 18"}
        };

        public static String answer3[]={
                "5 + 3",
                "7 + 5",
                "6 + 4",
                "9 + 6",
                "10 + 8",
                "12 + 8",
                "6 + 8",
                "9 + 7",
                "15 + 10",
                "18 + 12"

        };
    }
```

Ordering Number Activity

```
    public class OrderNumActivity extends AppCompatActivity implements
View.OnClickListener {
        TextView level,questionNumber, timer, question;
        Button opt1, opt2, opt3, submit;

        int questionIndex = 0;
        private int score = 0;
        int totalQuestion = OrderNumQues.question2.length;

        String selectedAns = "";


        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_ordering_number);

            //Connect using an XML tag that has an ID attribute.
            level = findViewById(R.id.level2);
            questionNumber = findViewById(R.id.questionindex2);
            timer = findViewById(R.id.timer2);


            question = findViewById(R.id.question2);
            opt1 = findViewById(R.id.Option1);
```

```
        opt2 = findViewById(R.id.Option2);
        opt3 = findViewById(R.id.Option3);
        //opt4 = findViewById(R.id.Option4);
        submit = findViewById(R.id.submit2);

        opt1.setOnClickListener(this);
        opt2.setOnClickListener(this);
        opt3.setOnClickListener(this);
        //opt4.setOnClickListener(this);
        submit.setOnClickListener(this);
        questionNumber.setText("Total questions: " + totalQuestion);
        submit.setEnabled(false);//Initially, the submission feature should be disabled.


        //set level text
        level.setText("Ordering Number");

        //load question
        loadNewQuestion();
    }
    public static ArrayList getRandomNonRepeatingIntegers(int size, int min, int max) {
        ArrayList numbers = new ArrayList();
        Random random = new Random();
        while (numbers.size() < size) {
            int randomNumber = random.nextInt((max - min) + 1) + min;
            if (!numbers.contains(randomNumber)) {
                numbers.add(randomNumber);
            }
        }
        return numbers;
    }
    @Override
    public void onClick(View v) {
        Button clickButton = (Button) v;

        if (clickButton.equals(submit)){
            if (selectedAns.equals(OrderNumQues.answer2[(int)
List.get(questionIndex)])){
                score++;
            }

            questionIndex++;
            loadNewQuestion();
        }
        else {
```

```
            selectedAns = clickButton.getText().toString();
            if (selectedAns == null) {
               submit.setEnabled(false); //If the selectedAnswer is null, then the submit
button should be disabled
            } else {
               submit.setEnabled(true); //If the the selectedAnswer is not null, then the
submit button be enable
            }
            showAns();

         }
      }

      ArrayList List =
getRandomNonRepeatingIntegers(OrderNumQues.question2.length, 0,
(OrderNumQues.question2.length-1));
      void loadNewQuestion(){

         if (questionIndex == totalQuestion) {
            finishQuiz();
            return;
         }
         submit.setEnabled(false);
         opt1.setEnabled(true);
         opt2.setEnabled(true);
         opt3.setEnabled(true);


         opt1.setBackgroundColor(Color.parseColor("#d3a04f"));
         opt2.setBackgroundColor(Color.parseColor("#d3a04f"));
         opt3.setBackgroundColor(Color.parseColor("#d3a04f"));



         questionNumber.setText("Question : "+(questionIndex+1)+" / "+totalQuestion );
         question.setText(OrderNumQues.question2[(int) List.get(questionIndex)]);
         opt1.setText(OrderNumQues.choice2[(int) List.get(questionIndex)][0]);
         opt2.setText(OrderNumQues.choice2[(int) List.get(questionIndex)][1]);
         opt3.setText(OrderNumQues.choice2[(int) List.get(questionIndex)][2]);

      }

      //finish quiz function
      void finishQuiz(){

         Intent intent = new Intent(OrderNumActivity.this,ResultScore.class);
```

```
        intent.putExtra("score",score);
        startActivity(intent);
        finish();
    }

    void showAns(){
        String b1=opt1.getText().toString();
        String b2=opt2.getText().toString();
        String b3=opt3.getText().toString();


        opt1.setEnabled(false);
        opt2.setEnabled(false);
        opt3.setEnabled(false);



        //opt1
        if (b1.equals(OrderNumQues.answer2[(int) List.get(questionIndex)])){
            opt1.setBackgroundColor(Color.parseColor("#66ff33"));
        }
        else {opt1.setBackgroundColor(Color.parseColor("#d80303"));}

        //opt2
        if (b2.equals(OrderNumQues.answer2[(int) List.get(questionIndex)])){
            opt2.setBackgroundColor(Color.parseColor("#66ff33"));
        }
        else {opt2.setBackgroundColor(Color.parseColor("#d80303"));}

        //opt3
        if (b3.equals(OrderNumQues.answer2[(int) List.get(questionIndex)])){
            opt3.setBackgroundColor(Color.parseColor("#66ff33"));
        }
        else {opt3.setBackgroundColor(Color.parseColor("#d80303"));}

    }
}
```

Ordering Number Questions

```
public class OrderNumQues {
    public static String question2[]={
            "Arrange the numbers 5, 2, 7, 1 in ascending order.",
            "Arrange the numbers 10, 15, 8, 12 in descending order.",
            "Arrange the numbers 3, 6, 2, 9 in ascending order.",
            "Arrange the numbers 20, 18, 25, 15 in descending order.",
```

```
            "Arrange the numbers 1, 3, 2, 4 in ascending order.",
            "Put the numbers 50, 40, 60, 30 in descending order.",
            "Arrange the numbers 12, 8, 15, 10 in ascending order.",
            "Arrange the numbers 100, 75, 125, 50 in descending order.",
            "Arrange the numbers 7, 4, 9, 5 in ascending order.",
            "the numbers 200, 150, 250, 100 in descending order."
    };

    public static String choice2[][]={
            {"1, 2, 7, 5 ","1, 2, 5, 7 ","7, 5, 2, 1 "},
            {"15, 12, 10, 8","8, 10, 12, 15","15, 10, 12, 8"},
            {"6, 3, 2, 9","2, 9, 6, 3","2, 3, 6, 9"},
            {"25, 18, 20, 15","25, 20, 18, 15","20, 25 , 18, 15"},
            {" 1, 3, 2, 4"," 1, 2, 3, 4"," 3, 4, 1, 2"},
            {"60, 30, 40, 50","60, 40, 50, 30","60, 50, 40, 30"},
            {"8, 15, 12, 10","8, 10, 12, 15","15, 10, 12, 8"},
            {"125, 100, 75, 50","100, 125, 75, 50","100, 75, 125, 50"},
            {"4, 5, 7, 9","4, 7, 5, 9","9, 7, 5, 4"},
            {"250, 200, 100, 150","200, 250, 150, 100","250, 200, 150, 100"},
    };

    public static String answer2[]={
            "1, 2, 5, 7 ",
            "15, 12, 10, 8",
            "2, 3, 6, 9",
            "25, 20, 18, 15",
            " 1, 2, 3, 4",
            "60, 50, 40, 30",
            "8, 10, 12, 15",
            "125, 100, 75, 50",
            "4, 5, 7, 9",
            "250, 200, 150, 100"
    };
}
```

Result Score

```
public class ResultScore extends AppCompatActivity implements
View.OnClickListener {
    TextView result, scoreGot;
    Button home, exit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_result_score);
```

```
        result = findViewById(R.id.Result);
        scoreGot = findViewById(R.id.scoreGot);
        home = findViewById(R.id.homebtn);
        exit = findViewById(R.id.exit);

        home.setOnClickListener(this);//What the fuck are you talking about
Bro!????????
        exit.setOnClickListener(this);

        Intent intent = getIntent();
        int score = intent.getIntExtra("score", 0);

        //Comparing Number result

        scoreGot.setText(String.format("Your Score is " + score + " out of " +
(ComNumQues.question1.length)));
        scoreGot.setText(String.format("Your Score is " + score + " out of " +
(OrderNumQues.question2.length)));
        scoreGot.setText(String.format("Your Score is " + score + " out of " +
(ComposingNumQues.question3.length)));
    }

    @Override
    public void onClick(View v) {

        Button clickButton = (Button) v;

        if (clickButton.equals(home)) {
            Intent intent = new Intent(ResultScore.this, ChooseTopic.class);
            startActivity(intent);
        } else {
            finishAffinity();
        }
    }

}
```