Inline Document: The document describes the code written to develop etaMath application.

1.) **Variables :** The variables initialised and defined for use in the source code are below:-

```
	public static Object value, value1, value2, value3, value4, value5, value6,
value7, value8, value9, value10,
			value11;
	public static Object value12, value13, value14, value15, value16, value17,
value18, value19;
	public static int option;
	public static int x = 10;
	public static String Name;
	public static int Passcount, Failcount = 0;
```

2.) **Constructor:** A constructor is used like below for the usage of Arraylist to store the Name of Person and their HighScores in the List.

```
public class Data {
		// global variables of the class
		String nameofPerson;
		int HighestScore;

		// constructor has type of data that is required
		Data(String nameofPerson, int HighestScore) {
	   // initialize the input variable from main

	   // function to the global variable of the class

			this.nameofPerson = nameofPerson;
			this.HighestScore = HighestScore;
		}
	}
```

3.) **Methods Used for option 2 (Enter '2' to SHOW the Highscores List):-**

```
private void addValues(String[] nameofPerson, int[] HighestScore) {
		ArrayList<Data> list = new ArrayList<>();

		for (int i = 0; i < x; i++) {
			list.add(new Data(nameofPerson[i], HighestScore[i]));
		}
		printValues(list);
	}

	private void printValues(ArrayList<Data> list) {

		for (int i = 0; i < x; i++) {
			Data data = list.get(i);
			System.out.println(data.nameofPerson + " " data.HighestScore);
		}
```

4.) **Methods Used for option 1 (Enter '1' to START the Game):-**

All the methods are used in the Main method

```
public static void main(String[] args) throws Exception {
MathsTasks.getInput_A();
                    MathsTasks.getInput_B();
                    MathsTasks.getInput_C();
                    MathsTasks.getInput_D();
                    MathsTasks.getInput_E();
                    MathsTasks.getInput_F();
                    MathsTasks.getInput_G();
                    MathsTasks.getInput_H();
                    MathsTasks.getInput_I();
                    MathsTasks.getInput_J();
                    MathsTasks.getInput_K();
                    MathsTasks.getInput_L();
                    MathsTasks.getInput_M();
                    MathsTasks.getInput_N();
                    MathsTasks.getInput_O();
                    MathsTasks.getInput_P();
                    MathsTasks.getInput_Q();
                    MathsTasks.getInput_R();
                    MathsTasks.getInput_S();
                    MathsTasks.getInput_T();


}
```

5.) The methods below are called from the main Method: -

The Methods cover all the 20 Tasks performing Addition, Subtraction, Multiplication and Division of two Integers/Numbers. These methods below also include the logic for **the User to input/Enter Answer within 10 Seconds.** If user doesn't enter the answer within 10 seconds time frame a warning is displayed saying **"...Sorry, you didn't responded fast enough".**

```
      // Task 1
      final static TimerTask task = new TimerTask() {
            public void run() {
                  if (value == null)
                        System.out.println("...Sorry, You didn't responded fast
enough");

            }
      };

      public static void getInput_A() throws Exception {
            Timer timer = new Timer();
            timer.schedule(task, 10 * 1000);

            System.out.println("Input/Enter your answer within 10 seconds: ");
            System.out.println("14 + 7 = ?");

            BufferedReader in = new BufferedReader(new
InputStreamReader(System.in));
            value = in.readLine();
            timer.cancel();

            // Addition of Two Numbers - Task 1
            if (value.equals("21")) {
                  System.out.println("Answer is Correct");
                  Passcount = Passcount + 1;
```

```
		} else {
			System.out.println("Answer is Wrong");
			Failcount = Failcount + 1;
		}
	}

	// Task 2
	final static TimerTask task1 = new TimerTask() {
		public void run() {
			if (value1 == null)
				System.out.println("...Sorry, You didn't responded fast
enough");

		}
	};

	public static void getInput_B() throws Exception {
		Timer timer1 = new Timer();
		timer1.schedule(task1, 10 * 1000);

		System.out.println("Input/Enter your answer within 10 seconds: ");
		System.out.println("50 - 14 = ?");

		BufferedReader Reader = new BufferedReader(new
InputStreamReader(System.in));
		value1 = Reader.readLine();
		timer1.cancel();

		// Substraction of Two Numbers - Task 2
		if (value1.equals("36")) {
			System.out.println("Answer is Correct");
			Passcount = Passcount + 1;
		} else {
			System.out.println("Answer is Wrong");
			Failcount = Failcount + 1;
		}
	}

	// Task 3
	final static TimerTask task2 = new TimerTask() {
		public void run() {
			if (value2 == null)
				System.out.println("...Sorry, You didn't responded fast
enough");

		}
	};

	public static void getInput_C() throws Exception {
		Timer timer2 = new Timer();
		timer2.schedule(task2, 10 * 1000);

		System.out.println("Input/Enter your answer within 10 seconds: ");
		System.out.println("14 + 36 = ?");

		BufferedReader Reader1 = new BufferedReader(new
InputStreamReader(System.in));
		value2 = Reader1.readLine();
		timer2.cancel();
```

```
		// Addition of Two Numbers - Task 3
		if (value2.equals("50")) {
			System.out.println("Answer is Correct");
			Passcount = Passcount + 1;
		} else {
			System.out.println("Answer is Wrong");
			Failcount = Failcount + 1;
		}

	}

	// Task 4
	final static TimerTask task3 = new TimerTask() {
		public void run() {
			if (value3 == null)
				System.out.println("...Sorry, You didn't responded fast
enough");

		}
	};

	public static void getInput_D() throws Exception {
		Timer timer2 = new Timer();
		timer2.schedule(task3, 10 * 1000);

		System.out.println("Input/Enter your answer within 10 seconds: ");
		System.out.println("8 * 7 = ?");

		BufferedReader Reader2 = new BufferedReader(new
InputStreamReader(System.in));
		value2 = Reader2.readLine();
		timer2.cancel();

		// Multiplication of Two Numbers -Task 4
		if (value2.equals("56")) {
			System.out.println("Answer is Correct");
			Passcount = Passcount + 1;
		} else {
			System.out.println("Answer is Wrong");
			Failcount = Failcount + 1;
		}

	}

	// Task 5
	final static TimerTask task4 = new TimerTask() {
		public void run() {
			if (value4 == null)
				System.out.println("...Sorry, You didn't responded fast
enough");

		}
	};

	public static void getInput_E() throws Exception {
		Timer timer2 = new Timer();
		timer2.schedule(task4, 10 * 1000);
```

```
            System.out.println("Input/Enter your answer within 10 seconds: ");
            System.out.println("55 / 11 = ?");

            BufferedReader Reader3 = new BufferedReader(new
InputStreamReader(System.in));
            value4 = Reader3.readLine();
            timer2.cancel();

            // Division of Two Numbers - Task 5
            if (value4.equals("5")) {
                  System.out.println("Answer is Correct");
                  Passcount = Passcount + 1;
            } else {
                  System.out.println("Answer is Wrong");
                  Failcount = Failcount + 1;
            }

      }

      // Task 6
      final static TimerTask task5 = new TimerTask() {
            public void run() {
                  if (value5 == null)
                        System.out.println("...Sorry, You didn't responded fast
enough");

            }
      };

      public static void getInput_F() throws Exception {
            Timer timer3 = new Timer();
            timer3.schedule(task5, 10 * 1000);

            System.out.println("Input/Enter your answer within 10 seconds: ");
            System.out.println("9 * 4 = ?");

            BufferedReader Reader4 = new BufferedReader(new
InputStreamReader(System.in));
            value5 = Reader4.readLine();
            timer3.cancel();

            // Division of Two Numbers - Task 6
            if (value5.equals("36")) {
                  System.out.println("Answer is Correct");
                  Passcount = Passcount + 1;
            } else {
                  System.out.println("Answer is Wrong");
                  Failcount = Failcount + 1;
            }

      }

      // Task 7
      final static TimerTask task6 = new TimerTask() {
            public void run() {
                  if (value6 == null)
                        System.out.println("...Sorry, You didn't responded fast
enough");
```

```
		}
	};

	public static void getInput_G() throws Exception {
		Timer timer3 = new Timer();
		timer3.schedule(task6, 10 * 1000);

		System.out.println("Input/Enter your answer within 10 seconds: ");
		System.out.println("35 / 5 = ?");

		BufferedReader Reader5 = new BufferedReader(new
InputStreamReader(System.in));
		value6 = Reader5.readLine();
		timer3.cancel();

		// Division of Two Numbers - Task 7
		if (value6.equals("7")) {
			System.out.println("Answer is Correct");
			Passcount = Passcount + 1;
		} else {
			System.out.println("Answer is Wrong");
			Failcount = Failcount + 1;
		}

	}

	// Task 8
	final static TimerTask task7 = new TimerTask() {
		public void run() {
			if (value7 == null)
				System.out.println("...Sorry, You didn't responded fast
enough");

		}
	};

	public static void getInput_H() throws Exception {
		Timer timer4 = new Timer();
		timer4.schedule(task7, 10 * 1000);

		System.out.println("Input/Enter your answer within 10 seconds: ");
		System.out.println("7 * 5 = ?");

		BufferedReader Reader6 = new BufferedReader(new
InputStreamReader(System.in));
		value7 = Reader6.readLine();
		timer4.cancel();

		// Multiplication of Two Numbers - Task 8
		if (value7.equals("35")) {
			System.out.println("Answer is Correct");
			Passcount = Passcount + 1;
		} else {
			System.out.println("Answer is Wrong");
			Failcount = Failcount + 1;
		}

	}
```

```
	// Task 9
	final static TimerTask task8 = new TimerTask() {
		public void run() {
			if (value8 == null)
				System.out.println("...Sorry, You didn't responded fast
enough");

		}
	};

	public static void getInput_I() throws Exception {
		Timer timer5 = new Timer();
		timer5.schedule(task8, 10 * 1000);

		System.out.println("Input/Enter your answer within 10 seconds: ");
		System.out.println("40 - 10 = ?");

		BufferedReader Reader7 = new BufferedReader(new
InputStreamReader(System.in));
		value8 = Reader7.readLine();
		timer5.cancel();

		// Substraction of Two Numbers - Task 9
		if (value8.equals("30")) {
			System.out.println("Answer is Correct");
			Passcount = Passcount + 1;
		} else {
			System.out.println("Answer is Wrong");
			Failcount = Failcount + 1;
		}

	}

	// Task 10
	final static TimerTask task9 = new TimerTask() {
		public void run() {
			if (value9 == null)
				System.out.println("...Sorry, You didn't responded fast
enough");

		}
	};

	public static void getInput_J() throws Exception {
		Timer timer6 = new Timer();
		timer6.schedule(task9, 10 * 1000);

		System.out.println("Input/Enter your answer within 10 seconds: ");
		System.out.println("10 + 30 = ?");

		BufferedReader Reader8 = new BufferedReader(new
InputStreamReader(System.in));
		value9 = Reader8.readLine();
		timer6.cancel();

		// Addition of Two Numbers - Task 10
		if (value9.equals("40")) {
			System.out.println("Answer is Correct");
			Passcount = Passcount + 1;
```

```
		} else {
			System.out.println("Answer is Wrong");
			Failcount = Failcount + 1;
		}

	}

	// Task 11
	final static TimerTask task10 = new TimerTask() {
		public void run() {
			if (value10 == null)
				System.out.println("...Sorry, You didn't responded fast
enough");

		}
	};

	public static void getInput_K() throws Exception {
		Timer timer7 = new Timer();
		timer7.schedule(task10, 10 * 1000);

		System.out.println("Input/Enter your answer within 10 seconds: ");
		System.out.println("85 - 25 = ?");

		BufferedReader Reader8 = new BufferedReader(new
InputStreamReader(System.in));
		value10 = Reader8.readLine();
		timer7.cancel();

		// Substraction of Two Numbers - Task 11
		if (value10.equals("60")) {
			System.out.println("Answer is Correct");
			Passcount = Passcount + 1;
		} else {
			System.out.println("Answer is Wrong");
			Failcount = Failcount + 1;
		}

	}

	// Task 12
	final static TimerTask task11 = new TimerTask() {
		public void run() {
			if (value11 == null)
				System.out.println("...Sorry, You didn't responded fast
enough");

		}
	};

	public static void getInput_L() throws Exception {
		Timer timer8 = new Timer();
		timer8.schedule(task11, 10 * 1000);

		System.out.println("Input/Enter your answer within 10 seconds: ");
		System.out.println("25 + 60 = ?");

		BufferedReader Reader9 = new BufferedReader(new
InputStreamReader(System.in));
```

```
			value11 = Reader9.readLine();
			timer8.cancel();

			// Addition of Two Numbers - Task 12
			if (value11.equals("85")) {
				System.out.println("Answer is Correct");
				Passcount = Passcount + 1;
			} else {
				System.out.println("Answer is Wrong");
				Failcount = Failcount + 1;

			}

		}

		// Task 13
		final static TimerTask task12 = new TimerTask() {
			public void run() {
				if (value12 == null)
					System.out.println("...Sorry, You didn't responded fast
enough");

			}
		};

		public static void getInput_M() throws Exception {
			Timer timer9 = new Timer();
			timer9.schedule(task12, 10 * 1000);

			System.out.println("Input/Enter your answer within 10 seconds: ");
			System.out.println("66 / 6 = ?");

			BufferedReader Reader9 = new BufferedReader(new
InputStreamReader(System.in));
			value12 = Reader9.readLine();
			timer9.cancel();

			// Division of Two Numbers - Task 13
			// value = in.readLine();
			if (value12.equals("11")) {
				System.out.println("Answer is Correct");
				Passcount = Passcount + 1;
			} else {
				System.out.println("Answer is Wrong");
				Failcount = Failcount + 1;
			}
		}

		// Task 14
		final static TimerTask task13 = new TimerTask() {
			public void run() {
				if (value13 == null)
					System.out.println("...Sorry, You didn't responded fast
enough");

			}
		};

		public static void getInput_N() throws Exception {
```

```
		Timer timer10 = new Timer();
		timer10.schedule(task13, 10 * 1000);

		System.out.println("Input/Enter your answer within 10 seconds: ");
		System.out.println("11 * 6 = ?");

		BufferedReader Reader10 = new BufferedReader(new
InputStreamReader(System.in));
		value13 = Reader10.readLine();
		timer10.cancel();

		// Multiplication of Two Numbers - Task 14
		if (value13.equals("66")) {
			System.out.println("Answer is Correct");
			Passcount = Passcount + 1;
		} else {
			System.out.println("Answer is Wrong");
			Failcount = Failcount + 1;
		}
	}

	// Task 15
	final static TimerTask task14 = new TimerTask() {
		public void run() {
			if (value14 == null)
				System.out.println("...Sorry, You didn't responded fast
enough");

		}
	};

	public static void getInput_O() throws Exception {
		Timer timer11 = new Timer();
		timer11.schedule(task14, 10 * 1000);

		System.out.println("Input/Enter your answer within 10 seconds: ");
		System.out.println("90 - 16 = ?");

		BufferedReader Reader10 = new BufferedReader(new
InputStreamReader(System.in));
		value14 = Reader10.readLine();
		timer11.cancel();

		// Substraction of Two Numbers - Task 15
		if (value14.equals("74")) {
			System.out.println("Answer is Correct");
			Passcount = Passcount + 1;
		} else {
			System.out.println("Answer is Wrong");
			Failcount = Failcount + 1;
		}
	}

	// Task 16
	final static TimerTask task15 = new TimerTask() {
		public void run() {
			if (value15 == null)
				System.out.println("...Sorry, You didn't responded fast
enough");
```

```
		}
	};

	public static void getInput_P() throws Exception {
		Timer timer12 = new Timer();
		timer12.schedule(task15, 10 * 1000);

		System.out.println("Input/Enter your answer within 10 seconds: ");
		System.out.println("16 + 74 = ?");

		BufferedReader Reader10 = new BufferedReader(new
InputStreamReader(System.in));
		value15 = Reader10.readLine();
		timer12.cancel();

		// Addition of Two Numbers - Task 16
		if (value15.equals("90")) {
			System.out.println("Answer is Correct");
			Passcount = Passcount + 1;
		} else {
			System.out.println("Answer is Wrong");
			Failcount = Failcount + 1;
		}
	}

	// Task 17
	final static TimerTask task16 = new TimerTask() {
		public void run() {
			if (value16 == null)
				System.out.println("...Sorry, You didn't responded fast
enough");

		}
	};

	public static void getInput_Q() throws Exception {
		Timer timer13 = new Timer();
		timer13.schedule(task16, 10 * 1000);

		System.out.println("Input/Enter your answer within 10 seconds: ");
		System.out.println("36 / 12 = ?");

		BufferedReader Reader11 = new BufferedReader(new
InputStreamReader(System.in));
		value16 = Reader11.readLine();
		timer13.cancel();

		// Division of Two Numbers - Task 17
		if (value16.equals("3")) {
			System.out.println("Answer is Correct");
			Passcount = Passcount + 1;
		} else {
			System.out.println("Answer is Wrong");
			Failcount = Failcount + 1;
		}

	}
```

```
	// Task 18
	final static TimerTask task17 = new TimerTask() {
		public void run() {
			if (value17 == null)
				System.out.println("...Sorry, You didn't responded fast
enough");

		}
	};

	public static void getInput_R() throws Exception {
		Timer timer14 = new Timer();
		timer14.schedule(task17, 10 * 1000);

		System.out.println("Input/Enter your answer within 10 seconds: ");
		System.out.println("3 * 12 = ?");

		BufferedReader Reader12 = new BufferedReader(new
InputStreamReader(System.in));
		value17 = Reader12.readLine();
		timer14.cancel();

		// Multiplication of Two Numbers - Task 18
		if (value17.equals("36")) {
			System.out.println("Answer is Correct");
			Passcount = Passcount + 1;
		} else {
			System.out.println("Answer is Wrong");
			Failcount = Failcount + 1;
		}
	}

	// Task 19
	final static TimerTask task18 = new TimerTask() {
		public void run() {
			if (value18 == null)
				System.out.println("...Sorry, You didn't responded fast
enough");

		}
	};

	public static void getInput_S() throws Exception {
		Timer timer15 = new Timer();
		timer15.schedule(task18, 10 * 1000);

		System.out.println("Input/Enter your answer within 10 seconds: ");
		System.out.println("29 - 7 = ?");

		BufferedReader Reader12 = new BufferedReader(new
InputStreamReader(System.in));
		value18 = Reader12.readLine();
		timer15.cancel();

		// Substraction of Two Numbers - Task 19
		if (value18.equals("22")) {
			System.out.println("Answer is Correct");
			Passcount = Passcount + 1;
		} else {
```

```
                        System.out.println("Answer is Wrong");
                        Failcount = Failcount + 1;

                }

        }

        // Task 20
        final static TimerTask task19 = new TimerTask() {
                public void run() {
                        if (value19 == null)
                                System.out.println("...Sorry, You didn't responded fast
enough");

                }
        };

        public static void getInput_T() throws Exception {
                Timer timer16 = new Timer();
                timer16.schedule(task19, 10 * 1000);

                System.out.println("Input/Enter your answer within 10 seconds: ");
                System.out.println("7 + 22 = ?");

                BufferedReader Reader12 = new BufferedReader(new
InputStreamReader(System.in));
                value19 = Reader12.readLine();
                timer16.cancel();

                // Addition of Two Numbers - Task 20
                if (value19.equals("29")) {
                        System.out.println("Answer is Correct");
                        Passcount = Passcount + 1;
                } else {
                        System.out.println("Answer is Wrong");
                        Failcount = Failcount + 1;
                }
        }
```

6.) **Highscore of the use**r answering all the questions correct and wrong are handled above in the code with the **variable 'Passcount' and 'Failcount' and is** displayed on the console as per the require ment**.**

```
System.out.println("Great! Within" + " " + Seconds + " " + "seconds," + "you
solved" + " " + Passcount + " "
                                        + "Tasks correctly!");
```

7.) **Fail count i.e the number of wrong answers** is also mentioned but commented in the code

```
// System.out.println("You have No: of wrong Answers : " +Failcount);
```

8.) **The Highscore shall be ordered** by correct answers is also handled in the code above with **'Passcount'** Variable.

```
if ((Passcount >= 16 && Passcount <= 20)) {
```

```
                        System.out.println("You made it to the Highest score!"
+ " " + "(Place" + " " + Passcount + ")");}
```

9.) **The Highscore shall be ordered** by **'used time'** in case correct answers where equal. The time is handled appropriately for printing the time taken for the game on the console with the variable **'Seconds'**

long StartTime = System.currentTimeMillis();

long EndTime = System.currentTimeMillis();

long Totaltime = EndTime - StartTime;

long Seconds = (Totaltime / 1000);

Printed on the console: - Great! Within 114 seconds, you solved 16 tasks correctly!

```
        System.out.println("Great! Within" + " " + Seconds + " " + "seconds," +
"you solved" + " " + Passcount + " "
                                + "Tasks correctly!");
```

10.) **When selecting option 2 Displays the list of Highest scores with the Name of the Person and his score:-**

```
        System.out.println(
                                "*******************************   HighScores
List    ****************************************");
```

```
Welcome to etaMath!
Enter '1' to START the Game
Enter '2' to SHOW the Highscores List
Enter Option '1' OR '2' :2
*******************************   HighScores List
****************************************
Michael 19
Peter 18
Andrew 17
Jay 20
Scott 16
Nikos 19
Paula 17
Smith 18
David 19
Simon 20
```

11.) **Finally the game is terminated:**

```
        Scanner inputName = new Scanner(System.in);
                    System.out.print("What is your Name? :");
                    Name = inputName.nextLine();
                    System.out.println("Thank you for playing the Game" + " " +
Name + "." + "The Game is now terminated.");
                    System.exit(0);
```