Formal Specification Description for the methods

Main Method

Since the main method takes the input from the user, these values should not be empty or null and the value of pi should not exceed 3.14(actual value of pi)

Formal Verification code for the main method:

```
/*@ requires radius>=0;
  @ ensures radius>= 0;
  @ ensures pilength > 0 && pilength <=3.14;
  @*/
```

AreaThread Constructor

AreaThread is a constructor, hence the values passed from the main program should not have any negative values and the initialization should happen properly. Hence the following verification code.

Formal Verification code for AreaThread Constructor:

```
/*@ requires radius>=0;
  @ ensures radius>= 0;
  @ ensures pi > 0 && pi <=3.14;
  @*/
```

Run method in AreaThread class

This run method calculates the area of the circle and hence the value of pi and radius should not be less than zero and area should be greater than or equal to zero.

Formal Verification code for run method:

```
/*@ requires pi_thread<=3.14 && pi_thread>=0
@ requires radius_thread>=0
@ ensures area >=0;
 @*/
```

Display method in AreaThread class

Since this method does only the task of display,there should be no change in the value of area which is calculated in the previous methods.

Formal Verification code for display method:

```
/*@ invariant area<0;
 @*/
```

CircumThread Constructor

CircumThread is a constructor, hence the values passed from the main program should not have any negative values and the initialization should happen properly. Hence the following verification code.

Formal Verification code for CircumThread constructor:

```
        /*@ requires radius>=0;
         @ ensures radius_thread>= 0;
         @ ensures pi_thread > 0 && pi_thread <=3.14;
         @*/
```

Run method in CircumThread class

This run method calculates the circumference of the circle and hence the value of pi and radius should not be less than zero and area should be greater than or equal to zero.

Formal Verification code for run method in CircumThread:

```
/*@ requires radius_thread>= 0;
         @ requires pi_thread > 0 && pi_thread <=3.14;
         @ ensures circum>=0;
         @*/
```

Display method in CircumThread class

Since this method does only the task of display, there should be no change in the value of circumference which is calculated in the previous methods.

Formal Verification code for display method:

```
/*@ invariant area<0;
 @*/
```

Run method in Timer thread class

Since this method is using a system variable to update the time variable,checking the value whether it is not null is important.Hence the below formal verification code.

Formal verification code for run method in Timer thread:

```
/*@ ensures timeStamp!=null;
 @*
```

IndiaTime constructor in IndiaTime thread class

Since the Timer thread calls the IndiaTime constructor with value including the current system time,the initialization has to be verified.

Formal Verification code for IndiaTime constructor:

```
/*@ ensures current_time!=null;
  @*/
```

Run method in IndiaTime thread class

Since the calculation for the India time is done in this method, the hours and minutes values have to be checked for their upper bound values.

Formal Verification code for run method in IndiaTime thread class:

```
/*@ requires current_time!=null;
  @ ensures h>= 0 && h<24;
  @ ensures m>0 && m<60;
  @ ensures current_time == \old(current_time);
  @*/
```