

**COE312 Software Design for Engineers**

**Homework 2: Threads and Threads Synchronization**

**Academic Integrity Pledge**

|  |
| --- |
| As a student of American University of Sharjah, I here by state that I will abide by the AUS Integrity Pledge that:   * I will hold myself accountable for all that I say and write. * I will hold myself responsible for the academic integrity of my work * I will not carry out unauthorized copying or printing of the work of others * I will not misrepresent my work nor give or receive unauthorized aid * I will behave in a manner that demonstrates concern for the personal dignity, rights and freedoms of all members of the community * I will respect university property and the property of others; and * I will not tolerate a lack of respect for these values.   **Student Name: Amir Mohideen**  **Student ID: 74559**  **Student Name: yazan**  **Student ID: 74595** |

**Note: You must submit the code files as well.**

**Github link:** https://github.com/COE312L/HW2

**Java code**

*Please add subtitles with the class names. Paste only the new code, and* ***not*** *screenshots of the code.*

//main that implements the 4 required threads along with shared object location

**import** java.util.\*;

**public** **class** main {

**public** **static** **void** main(String[] args) **throws** InterruptedException{

// **TODO** Auto-generated method stub

loc l = **new** loc();

//time t = new time();

mainthread main = **new** mainthread(l);

Watch w = **new** Watch();

Harry h = **new** Harry(l);

Hermione her = **new** Hermione(l);

}

}

//-----------------

//main thread class that has the entire main code of HW1 along with internal synchronization changes made to the code where location is used

**import** java.util.Scanner;

**public** **class** mainthread **implements** Runnable{

//int n;

loc loc;

Thread t;

mainthread(loc l1) {

**this**.loc = l1;

t = **new** Thread(**this**);

t.start();

}

**public** **void** run() {

//Initializations

Library l=**new** Library();

DiningHall d=**new** DiningHall();

PotionsClassroom p=**new** PotionsClassroom();

//we assume the user types all input in lowercase

// For password

Scanner in = **new** Scanner(System.***in***);

System.***out***.println("“Messrs Moony, Wormtail, Padfoot and Prongs Purveyors of Aids to magical Mischief-Makers are proud to present THE MARAUDER'S MAP”\n" +

"Say the magic phrase to reveal the map.\n");

String input = in.nextLine();

**while**(!input.contains("i solemnly swear that i am up for no good")) {

System.***out***.println("Wrong Password! Try Again:\n");

input = in.nextLine();

}

//while loop to continue asking user questions until user decides to exit by typing 'mischief managed'

loc.setState("Entrance"); //to enter switch case

**while**(!input.contains("mischief managed")) {

**switch**(loc.getState()) {

//contains 4 cases with 4 locations: Entrance, Dining Hall, Library, Potions Classroom

**case** "Entrance":

**while**(loc.getState().contains("Entrance")) {

System.***out***.println("The Entrance\n" +

"You arrive at the doors of Hogwarts. The door on the north wall leads to the dining hall,\n"+

"the door to the east leads to the Potions class, and the door to the west leads to the Library.");

System.***out***.println("Which direction would you like to walk to? (walk north/east/west)\n");

input = in.nextLine();

**if**(input.contains("north")) {

System.***out***.print("You have entered the dining hall. (look around/talk to character)\n");

loc.setState("Dining Hall");

**break**;

}

**else** **if**(input.contains("west")) {

System.***out***.print("You have entered the Library. (look around/talk to character)\n");

loc.setState("Library");

**break**;

}

**else** **if**(input.contains("east")) {

System.***out***.print("You have entered the potions class. (look around/talk to character)\n");

loc.setState("Potions Classroom");

**break**;

}

**else** **if** (input.contains("mischief managed")) {

**break**;

}

**else** {

System.***out***.print("Wrong input! Try again\n");

}

}

**case** "Dining Hall":

**while**(loc.getState().contains("Dining Hall")) {

input = in.nextLine();

**if**(input.contains("look")) {

d.look();

}

**else** **if**(input.contains("talk")) {

**if**(input.contains("dumbledore")) {

d.talk();

}

**else** {

System.***out***.print("This character is not here.\n");

}

}

**else** **if**(input.contains("use")) {

**if**(input.contains("hat")) {

d.use("sorting hat");

}

**else** **if**(input.contains("plate")) { //changed to plate to make a meaningful sentence in output

d.use("plate");}

**else** {

System.***out***.print("You can only use a sorting hat or plate.\n");

}

}

**else** **if**(input.contains("north")||input.contains("east")||input.contains("west")) {

System.***out***.print("You find that there is only one direction to walk, south\n");

}

**else** **if** (input.contains("south")) {

loc.setState("Entrance");

}

**else** **if** (input.contains("mischief managed")) {

**break**;

}

**else** {

System.***out***.print("Wrong input");

}

}

**break**;

**case** "Library":

**while**(loc.getState()=="Library") {

input = in.nextLine();

**if** (input.contains("look")) {

l.look();

}

**else** **if**(input.contains("talk")) {

**if**(input.contains("hermione")) {

l.talk();

}

**else** {

System.***out***.print("This character is not here!\n");}

}

**else** **if**(input.contains("use")) {

**if**(input.contains("book")) {

l.use("book");

}

**else** **if**(input.contains("quill")) {

l.use("quill");

}

**else** {

System.***out***.print("You can only use a book or quill.\n");}

}

**else** **if**(input.contains("north")||input.contains("west")||input.contains("east")) {

System.***out***.print("You find that there is only one direction to walk, south\n");

}**else** **if** (input.contains("south")) {

loc.setState("Entrance");

}

**else** **if** (input.contains("mischief managed")) {

**break**;

}

**else** {

System.***out***.print("Wrong input!\n");

}

}

**break**;

**case** "Potions Classroom":

**while**(loc.getState()=="Potions Classroom") {

input = in.nextLine();

**if** (input.contains("look")) {

p.look();

}

**else** **if**(input.contains("talk")) {

**if**(input.contains("snape")) {

p.talk();

}

**else** {

System.***out***.print("This character is not here!\n");

}

}

**else** **if**(input.contains("use")) {

**if**(input.contains("cauldron")) {

p.use("cauldron");

}

**else** **if**(input.contains("potion")){

p.use("potion");

}

**else** {

System.***out***.print("You can only use a cauldron or a potion bottle.\n");}

}

**else** **if**(input.contains("north")||input.contains("west")||input.contains("east")) {

System.***out***.print("You find that there is only one direction to walk, south\n");

}

**else** **if** (input.contains("south")) {

loc.setState("Entrance");

}

**else** **if**(input.equals("mischief managed")) {

**break**;

}

**else** {

System.***out***.print("Wrong input!\n");

}

}

**break**;

**default**: System.***out***.print("Incorrect input!\n");

}

}

in.close();

System.***out***.print("Hiding map contents...end\n");

}

}

//-------------------------------------

//class harry that shares object loc and contains thread runs function

**public** **class** Harry **implements** Runnable{

loc l;

Harry(loc l1) {

**this**.l = l1;

Thread t = **new** Thread(**this**);

t.start();

}

**public** **void** run() {

**while**(**true**)

{

**if**(l.getState()=="Library") {

System.***out***.println("Found Hermione!");

**break**;

}

}

}

}

//class Hermione that shares object loc and checks if location is library and contains thread function run

**public** **class** Hermione **implements** Runnable{

loc l;

Hermione(loc l1) {

Thread t = **new** Thread(**this**);

**this**.l=l1;

t.start();

}

**public** **void** run() {

**while**(**true**)

{

**if**(l.getState()=="Library")

{

System.***out***.println("Harry I am here!");

**break**;

}

}

}

}

//class loc that contains location and has internal synchronization and is shared by mainthread, harry and hermione

**public** **class** loc {

String location;

loc() {

location = "none";

}

**public** **synchronized** String getState() {

**return** location;

}

**public** **synchronized** **void** setState(String l) {

**this**.location = l;

}

}

//watch class that notifies the harry the amount of time left for him to find hermione and exit if times up

**import** java.util.concurrent.TimeUnit;

**public** **class** Watch **implements** Runnable{

//time tim;

Thread t;

Watch() {

//this.tim=t1;

t = **new** Thread(**this**);

t.start();

}

**public** **void** run() {

//long start = System.currentTimeMillis();

**try** {

TimeUnit.***SECONDS***.sleep(60);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

System.***out***.println( "1 minute passed...");

**try** {

TimeUnit.***SECONDS***.sleep(60);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

System.***out***.println( "2 minutes passed...");

**try** {

TimeUnit.***SECONDS***.sleep(60);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

System.***out***.println( "3 minutes passed...Hide map contents before someone sees you!\n");

**try** {

TimeUnit.***SECONDS***.sleep(10);

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

System.***out***.println( "You have been caught... and the Marauder’s map has been confiscated.");

System.*exit*(0);

}

}

**Sample Output**

1. Paste screenshot of output showing that Harry found Hermione before time is up and then hiding the map to end.

Graphical user interface, text, application, email

Description automatically generated

1. Paste screenshot of output showing that Harry got caught (time is up and he did not hide the map).

Text, letter

Description automatically generated