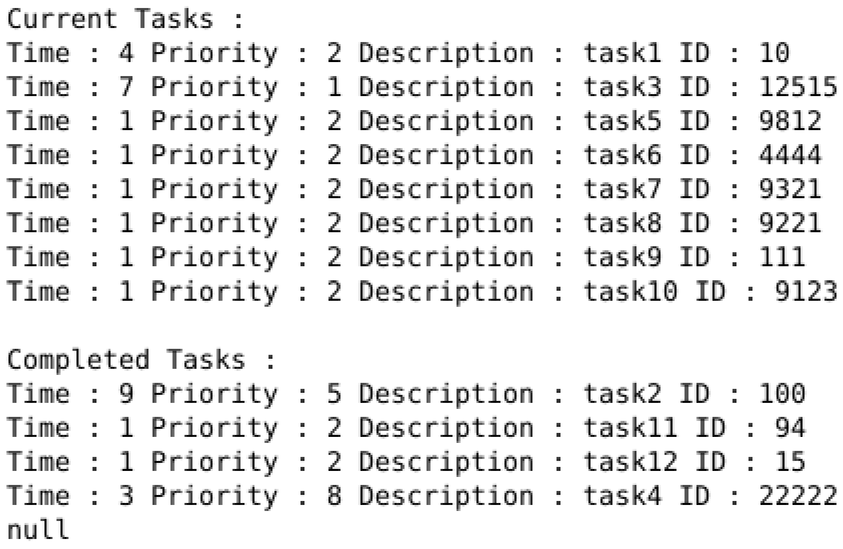
```
/* Havin Lim / Quynh Dao
Lab: 25 - Task ArrayList
October 31st 2022
Sources : None
Help obtained : None
We confirm that the above list of sources is complete AND that we have not talked to anyone
else (e.g., CSC207 students) about the solution to this problem
 */

// Add Method

/**
 * Method that adds the input value (Task) to a certain TaskArrayList
 * @return boolean
 */
public boolean add(Task value) {
        ensureCapacity(size+1);
        taskData[size] = value;
        size++;
        return true;
}

// ensureCapacity Method
/**
 * Method that ensures the arraylist has enough capacity for tasks to be added
 * @param capacity    int
 */
public void ensureCapacity(int capacity) {
        if (capacity > taskData.length) {
                int newCapacity = taskData.length + 1;
                if (capacity > newCapacity) {
                        newCapacity = capacity;
                }
                taskData = Arrays.copyOf(taskData, newCapacity);
        }
}
```

```
// removebyID Method

/**
 * Method that removes the task with the input ID and returns the removed Task
 * If the input ID is wrong (not in the TaskArrayList, it returns null
 * @param ID int
 * @return Task
 */
public Task removebyID(int ID) {
    for (int i = 0; i < size; i ++) {
        if(ID == taskData[i].getID()) {
            Task keep = taskData[i];
            for(int l = i; l < taskData.length - 1 ; l++) {
                taskData[l] = taskData[l+1];
            }
            taskData = Arrays.copyOf(taskData, taskData.length -1);
            size --;
            return keep;
        }
    }
    return null;
}

// TaskArrayListTester.java

package Lab25;

public class TaskArrayListTester {

    public static void main(String[] args) {

        TaskArrayList task = new TaskArrayList();
        TaskArrayList taskComplete = new TaskArrayList();

        Task task1 = new Task(4, 2, 10, "task1");
        Task task2 = new Task(9, 5, 100, "task2");
        Task task3 = new Task(7, 1, 12515, "task3");
        Task task4 = new Task(3, 8, 22222, "task4");
        Task task5 = new Task(1, 2, 9812, "task5");
        Task task6 = new Task(1, 2, 4444, "task6");
        Task task7 = new Task(1, 2, 9321, "task7");
        Task task8 = new Task(1, 2, 9221, "task8");
        Task task9 = new Task(1, 2, 111, "task9");
        Task task10 = new Task(1, 2, 9123, "task10");
        Task task11 = new Task(1, 2, 94, "task11");
        Task task12 = new Task(1, 2, 15, "task12");

        task.add(task1);
        task.add(task2);
        task.add(task3);
        task.add(task4);
        task.add(task5);
        task.add(task6);
        task.add(task7);
        task.add(task8);
        task.add(task9);
        task.add(task10);
        task.add(task11);
        task.add(task12);
        taskComplete.add(task.removebyID(100));
        taskComplete.add(task.removebyID(94));
        taskComplete.add(task.removebyID(15));
        taskComplete.add(task.removebyID(22222));
        taskComplete.add(task.removebyID(9999999));

        System.out.println("Current Tasks : ");
        task.printAll();
        System.out.println("");
        System.out.println("Completed Tasks : ");
        taskComplete.printAll();

    }


}
```

// Test Output

```
Current Tasks :
Time : 4 Priority : 2 Description : task1 ID : 10
Time : 7 Priority : 1 Description : task3 ID : 12515
Time : 1 Priority : 2 Description : task5 ID : 9812
Time : 1 Priority : 2 Description : task6 ID : 4444
Time : 1 Priority : 2 Description : task7 ID : 9321
Time : 1 Priority : 2 Description : task8 ID : 9221
Time : 1 Priority : 2 Description : task9 ID : 111
Time : 1 Priority : 2 Description : task10 ID : 9123

Completed Tasks :
Time : 9 Priority : 5 Description : task2 ID : 100
Time : 1 Priority : 2 Description : task11 ID : 94
Time : 1 Priority : 2 Description : task12 ID : 15
Time : 3 Priority : 8 Description : task4 ID : 22222
null
```

// Screenshot of the test output



```
Current Tasks :
Time : 4 Priority : 2 Description : task1 ID : 10
Time : 7 Priority : 1 Description : task3 ID : 12515
Time : 1 Priority : 2 Description : task5 ID : 9812
Time : 1 Priority : 2 Description : task6 ID : 4444
Time : 1 Priority : 2 Description : task7 ID : 9321
Time : 1 Priority : 2 Description : task8 ID : 9221
Time : 1 Priority : 2 Description : task9 ID : 111
Time : 1 Priority : 2 Description : task10 ID : 9123

Completed Tasks :
Time : 9 Priority : 5 Description : task2 ID : 100
Time : 1 Priority : 2 Description : task11 ID : 94
Time : 1 Priority : 2 Description : task12 ID : 15
Time : 3 Priority : 8 Description : task4 ID : 22222
null
```

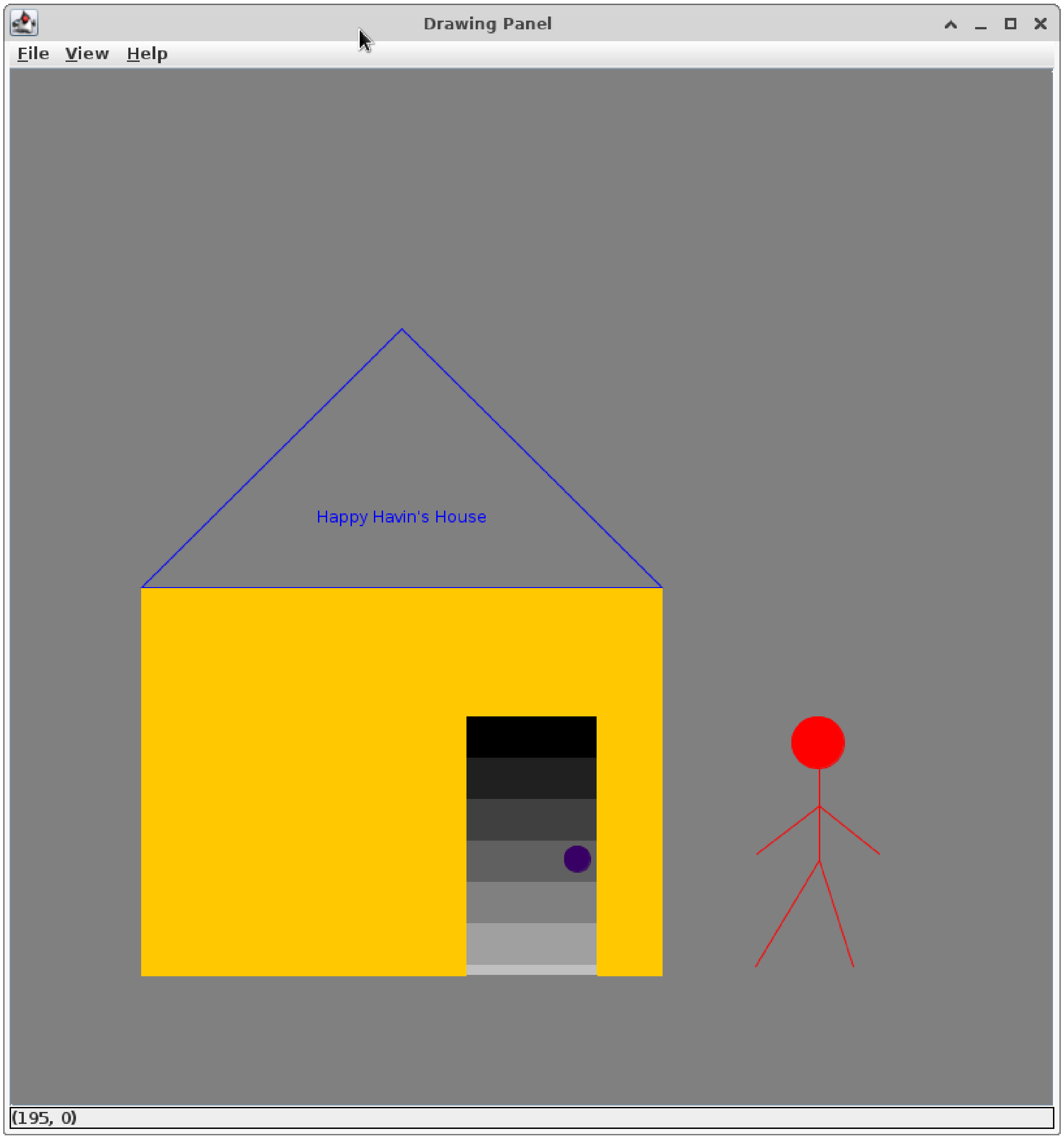
```
/* Havin Lim / Quynh Dao
Lab: 26 - LinkedList Implementation
November 2nd 2022
Sources : None
Help obtained : None
We confirm that the above list of sources is complete AND that we have not talked to anyone
else (e.g., CSC207 students) about the solution to this problem
 */


// Queue207Implementation.java

package Lab26;

import java.lang.NullPointerException;
import java.util.NoSuchElementException;

public class Queue207Implementation <AnyType> implements Queue207<AnyType> {

    int numberItems = 0;
    QueueNode<AnyType> head = null;
    QueueNode<AnyType> tail = null;


    @Override
    public boolean add(AnyType e) throws NullPointerException {
        QueueNode<AnyType> abc = new QueueNode<>(e, null);
        if (numberItems != 0) {
            QueueNode<AnyType> temp = tail;
            temp.setNext(abc);
            tail = abc;
        }
        else {
            head = abc;
            tail = abc;
        }
        numberItems++;
        return true;
    }
    @Override
    public AnyType remove() throws NoSuchElementException {
        if(head == null) {
            throw new NoSuchElementException();
        }
        else if (numberItems == 1) {
            AnyType removed = head.getData();
            head = null;
            tail = null;
            numberItems--;
            return removed;
        }
        else {
        AnyType removed = head.getData();
        head = head.getNext();
        numberItems--;
        return removed;
        }
    }
    @Override
    public AnyType element() throws NoSuchElementException {
        return head.getData();
    }
    @Override
    public int size() {
        return numberItems;
    }

}
```

```java
// Queue207ImplementationTester.java

package Lab26;

public class Queue207ImplementationTester {
        public static void main (String[] argrs) {
                Queue207Implementation<Integer> queue = new Queue207Implementation<>();
                System.out.println(queue.add(10));
                System.out.println(queue.add(9));
                System.out.println(queue.remove());
                System.out.println(queue.element());
                System.out.println(queue.size());
                System.out.println(queue.add(3));
                System.out.println(queue.size());
                System.out.println(queue.remove());
                System.out.println(queue.remove());
                System.out.println(queue.size());
                Queue207Implementation<String> queue2 = new Queue207Implementation<>();
                System.out.println(queue2.add("hehe"));
                System.out.println(queue2.add("hihi"));
                System.out.println(queue2.remove());
                System.out.println(queue2.element());
                System.out.println(queue2.size());
                System.out.println(queue2.add("havin"));
                System.out.println(queue2.size());
                System.out.println(queue2.remove());
                System.out.println(queue2.remove());
                System.out.println(queue2.size());
        }

}
```

```
// Output of the client program

true
true
10
9
1
true
2
9
3
0
true
true
hehe
hihi
1
true
2
hihi
havin
0
```

```
/* Havin Lim / Quynh Dao
Lab: 27 - Drawing and the Java GUI
November 4th 2022
Sources : None
Help obtained : None
We confirm that the above list of sources is complete AND that we have not talked to anyone
else (e.g., CSC207 students) about the solution to this problem
 */

// DrawingComplex.java

package Lab27;
import java.awt.*;

public class DrawingComplex {
        public static void main(String[] args) {
                DrawingPanel panel = new
DrawingPanel(Integer.parseInt(args[0]),Integer.parseInt(args[1]));
                panel.setBackground(Color.GRAY);
                Color violet = new Color(56,0,100);
                Graphics g = panel.getGraphics();

                g.setColor(Color.orange);
                g.drawRect(100, 400, 400, 300);
                g.fillRect(100, 400, 400, 300);
                g.setColor(Color.BLUE);
                g.drawPolygon(new int[] {100, 300, 500}, new int[] {400, 200, 400}, 3);
                g.drawString("Happy Havin's House", 235, 350);
                g.setColor(Color.RED);
                for(int i=0; i<8; i++) {
                        int shift = i * 256 / 8;
                        g.setColor(new Color(shift, shift, shift));
                        int yaxis = 500 + i * 256 / 8;
                        g.fillRect(350, yaxis , 100, 200);
                }

                g.setColor(Color.GRAY);
                g.drawRect(350, 700, 100, 200);
                g.fillRect(350, 700, 100, 200);
                g.setColor(violet);
        g.drawOval(425, 600, 20, 20);
        g.fillOval(425, 600, 20, 20);


        g.setColor(Color.RED);
        // Head
        g.drawOval(600, 500, 40, 40);
        g.fillOval(600, 500, 40, 40);

        //Body
        g.drawLine(621, 541, 621, 611);

        // Left Leg
        g.drawLine(621, 611, 572, 693);

        // Right Leg
        g.drawLine(621, 611, 647, 693);

        // Left Arm
        g.drawLine(621, 569, 573, 606);

        // Right Arm
        g.drawLine(621, 569, 667, 606);
        }
    }
```

// Output of the program


Drawing Panel
File View Help
Happy Havin's House
(195, 0)