Midterm 1 Question 1

Submit 1 file: PDF, PNG, or JPEG only.

5 marks.

Java to UML Class Diagram

Convert this Java code to a UML class diagram. This Java code is meant to represent a player character in a video game where you can use the jump button and directional pad to move the character around. If the player character gets a power up like a flying carpet they might get the `FlyingPowerupSuit` that enables their jumps to help them fly.

Draw a well-designed UML class diagram to represent this information. Provide the basic abstractions, attributes, methods, relationships, multiplicities, and navigabilities as appropriate. "..." and "// ..." means much code is omitted. Static variables can be underlined to indicate they are static.

```
class PlayerCharacter {
  private PowerupSuit myBehaviour;
  // delegate jump button movement
  public void jump() {
     myBehaviour.jump(this);
  }
  // delegate direction pad movement
  public void direction(Direction d) {
     myBehaviour.direction(this, d);
  }
  public void move(Direction d) {
    // ...
  }
  // Is the player on the ground?
  public boolean onGround() {
    // ...
  }
}

public class Direction {
  // constant represents UP
```

```
  final static Direction UP = new Direction(0,1);
  // ...
}

public interface PowerupSuit {
  // Jump button pressed
  public void jump(PlayerCharacter p);
  // Direction pad indicated
  public void direction(PlayerCharacter p, Direction d);
}
public class BasicPowerupSuit implements PowerupSuit {
  public void jump(PlayerCharacter p) {
    if (p.onGround()) {
       p.move(Direction.UP);
    }
  }
  public void direction(PlayerCharacter p, Direction d) {
     p.move(d);
  }
  // ...
}

public class FlyingPowerupSuit implements PowerupSuit {
  public void jump(PlayerCharacter p) {
     p.move( Direction.UP);
    p.move(Direction.UP);
  }
  public void direction(PlayerCharacter p, Direction d) {
     p.move(d);
  }
  // ...
}
```

Midterm 1 Question 2

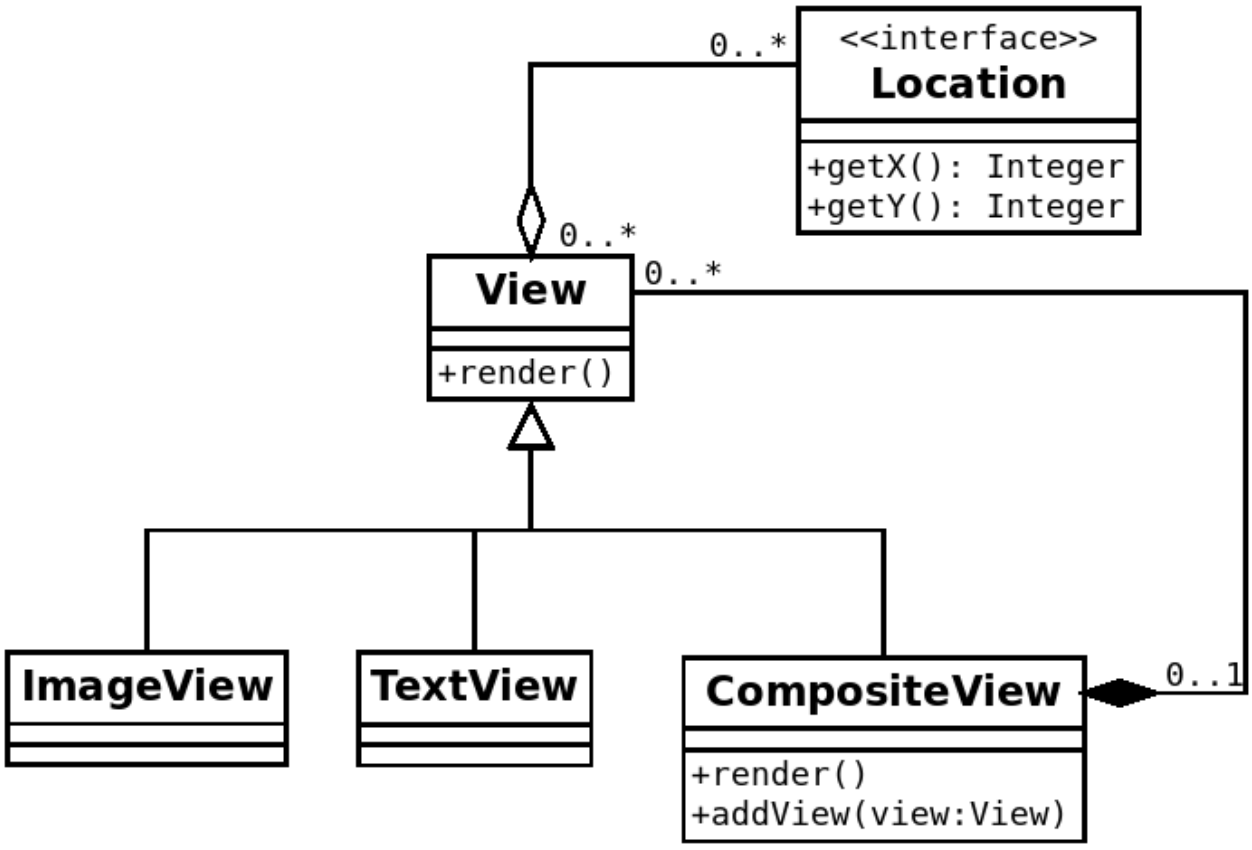
Plain text preferred: 1 .txt file or 1 .java file

Submit .txt, .java, a zip of .java, PDF, PNG, or JPEG only.

5 marks

UML Class Diagram to Java

Convert this UML class diagram of a Graphical User Interface Toolkit to skeletal Java Code. Include all attributes and obviously public methods. Make sure all required methods are implemented. Include all generalizations and necessary associations. You may use ellipses (... or // ...) to indicate there is some code in the method but it's a midterm and who has time to write that code!



ACCOMMODATIONS: If you need a textual version of this UML PDFs and SVGs of the diagram are available for you.

Midterm 1 Question 3

Submit 1 file: PDF, PNG, or JPEG only.

5 marks

Liskov Substitution Principle

This Java code is poor! It violates the Liskov Substitution principle!

Restructure this code as UML and draw a UML Class Diagram of restructured code that doesn't violate the Liskov Substitution principle.

Draw a well-designed UML class diagram to represent this information. Provide the basic abstractions, attributes, methods, relationships, multiplicities, and navigabilities as appropriate.

"..." and “// …” means much code is omitted.

This Java code is about reading CBZ files which are compressed comic books. They are a zip file of PNGs with ordered filenames.

Be sure to leave a note on the diagram highlighting how you address the violation

```
public interface FileContent {
  public Path getPath();
  public byte[] getData();
}

public class ZipFile {
  // Work on the zip file File zipFile
  protected File zipFile;
  // Represents a single zip file for reading and writing
  public ZipFile(File zipFile) {
    // ...
  }
  // Gets the file contents from the zipfile
  public FileContent getFileContent(Path path) {
    // ...
```

```
  }
  // return all paths
  public Path[] getPaths() {
    // ...
  }

  // Get all contents from all files within the zipfile
  public FileContent[] getFileContents() {
    // ...
  }
  // Adds a file to the zip file
  public void addFile(File f) {
    // ...
  }
  // saves all the files as a Zip
  public void save() {
    // ...
  }
  // ...
}

public class ComicBookCBZReader extends ZipFile {
  public ComicBookCBZReader(File f) {
    super(f);
    // ...
  }
  public Image[] getPages() {
    FileContent[] imageFiles = getFileContents();
    // foreach imageFile convert to Image
    // ...
  }
  public void addFile(File f) {}
  public void save() {
    // we just read, we don't need to save
  }
  // ...
}
```

Midterm 1 Question 1

Submit PDF, PNG, or JPEG only.

5 marks

Java to UML Class Diagram

Convert this Java code to a UML class diagram. This Java code is meant to represent some Zerg defence structures in Starcraft. Specifically, a Spore Colony is a Zerg means of anti-aerial defence. It's able to fling corrosive spores (up to 8 per colony) high into the atmosphere. The spores can damage enemy air units.

Draw a well-designed UML class diagram to represent this information. Provide the basic abstractions, attributes, methods, relationships, multiplicities, and navigabilities as appropriate. "..." means much code is omitted.

```
public abstract class Structure {
    // Some structures evolve from other structures,
    // some structures don't have anything to evolve from
    protected Structure evolvesFrom;
    protected int buildTime;

    public int getBuildTime() {
        return buildTime;
    }

}

public interface AirDefence {
    public int getAirDamage();
}

public abstract class Colony extends Structure {
    protected int defence;

    public abstract int getDefencePoints();
}

public class SporeColony extends Colony implements AirDefence {
    private List<Spore> spores = Arrays.asList(new Spore[8]);

    public int getAirDamage() {
        // ...
    }

    public int getDefencePoints() {
        return defence;
    }

    public int sporesLeft() {
        // ...
    }
}

public class Spore {
```

```
    // ...
}
```

Midterm 1 Question 2

Plain text preferred: .txt or .java

Submit .txt, .java, a zip of .java, PDF, PNG, or JPEG only.

5 marks

UML Class Diagram to Java

Convert this UML class diagram of a Spotify-like service to skeletal Java Code. Include all attributes and obviously public methods. Make sure all required methods are implemented. Include all generalizations and necessary associations. You may use ellipses (...) to indicate there is some code in the method but it's a midterm and who has time to write that code!

ACCOMMODATIONS: If you need a textual version of this UML Q2.pdf and Q2.svg are available for you.

Midterm 1 Question 3

Submit PDF, PNG, or JPEG only

5 marks

Liskov Substitution Principle

This Java code is poor! It violates the Liskov Substitution principle!

Restructure this code as UML and draw a UML Class Diagram of restructured code that doesn't violate the Liskov Substitution principle.

Draw a well-designed UML class diagram to represent this information. Provide the basic abstractions, attributes, methods, relationships, multiplicities, and navigabilities as appropriate. "..." means much code is omitted.

This Java code is about Canadian geese, their baby goslings and some of the things they can do.

Be sure to leave a note on the diagram highlighting how you address the violation

```
public class Goose {
    protected boolean inAir;
    private ArrayList<BabyGoose> goslings = new ArrayList<BabyGoose>();
    public Goose() {
        ...
    }
    public void takeOff() {
        this.inAir = true;
    }
    public void chase(Human human) {
        ...
          human.runAway();
    }
}

public class BabyGoose extends Goose {
    public BabyGoose() {
        super();
        this.inAir = false;
    }
    public void takeOff() {
        System.out.println("I’m still learning how to fly!");
    }
}

public interface Chaseable {
    public int getRunningAwaySpeed();
    public void runAway();
}

public class Human implements Chaseable {
    private int runningAwaySpeed = 2; // m/s
    public void runAway() {
        System.out.println("I’m running from an angry goose!");
    }
    public int getRunningAwaySpeed() {
```

```
        return runningAwaySpeed;
    }
}
```