This exam will cover material from Chapter 6, 7, and 8

Here are some general topics for the exam from those chapters:

- ~~**method overloading**~~
- ~~**calling a superclass constructor from subclass**~~
- **static and final**
- **java.util.List objects**
- **for-each loop**
- **intersecting objects (using getOneIntersectingObject method)**
- **using GreenfootImages and painting our own images**
- **for-loops**
- **animation (explosions and smoke examples from class)**
- ~~**using the mouse to capture user interactions**~~

(1) ~~What is overloading (specifically method overloading)? When is it used?~~

(2) When are static and final used in a variable declaration?

Static and final are used when you want to make the variable a constant. That is, you want the value of the variable never to change during the execution of the program.

(3) Write the code to see if an actor is colliding with another (an Obstacle). If there is a collision, the image of the Obstacle should change to "nothreat.png".

```
Actor a = getOneIntersectingObject(Obstacle.class);
if (a != null) {
   a.setImage(new GreenfootImage("nothreat.png"));
}
```

(4) Write the code to create a GreenfootImage with a blue background and five white squares at random locations. The squares should be sized 20x20.

```
GreenfootImage image = new GreenfootImage(1000, 1000);
image.setColor(java.awt.Color.BLUE);
image.fill();
image.setColor(java.awt.Color.WHITE);
image.fillRectangle(Greenfoot.getRandomNumber(getWidth()),
Greenfoot.getRandomNumber(getHeight()), 20, 20);
image.fillRectangle(Greenfoot.getRandomNumber(getWidth()),
Greenfoot.getRandomNumber(getHeight()), 20, 20);
image.fillRectangle(Greenfoot.getRandomNumber(getWidth()),
Greenfoot.getRandomNumber(getHeight()), 20, 20);
image.fillRectangle(Greenfoot.getRandomNumber(getWidth()),
Greenfoot.getRandomNumber(getHeight()), 20, 20);
image.fillRectangle(Greenfoot.getRandomNumber(getWidth()),
Greenfoot.getRandomNumber(getHeight()), 20, 20);

int count = 0;
while (count < 5) {
        image.fillRectangle(Greenfoot.getRandomNumber(getWidth()),
        Greenfoot.getRandomNumber(getHeight()), 20, 20);
        count = count + 1;
}

for(int count = 0; count < 5; count = count + 1) {
        image.fillRectangle(Greenfoot.getRandomNumber(getWidth()),
        Greenfoot.getRandomNumber(getHeight()), 20, 20);
}
```

(5) Use the following for-loop definition to answer parts a – g.

```
for(int count = 1; count < 9; count = count + 1) {
   addObject(new Question(), 34, 34);
}
```

a) What is the initial value of this loop's counter variable?
1

b) What is the value of this loop's counter variable when the loop is done executing?
9

c) Circle the part of the code above that is considered the loop body.

d) initialization

e) condition

f) increment

g) How many times would this loop execute?
8

(6) Which of the following would be the correct choice to fill in the blank in the code to make this loop execute 5 times?

```
for (int count = 1; ______________; count++) {
     //some code for loop
}
```

a) `count < 5`
b) `count <= 5`
c) `count < 6`
d) `count <= 6`

(7) Write the code that gets all the Flowers from the scenario and then moves each flower 5 pixels to its right.

```
java.util.Collection<Flower> flowers = getWorld().getObjects(Flower.class);

for (Flower f: flowers) {
        f.setLocation(f.getX() + 5, f.getY());
}
```

Look at the questions in the worksheet for Chapter 8.

~~**(8)** Write the code so that an actor will follow the movements of the mouse on the world.~~

~~**(9)** Reproduce the code in the grow() or shrink() methods that we discussed in class that goes through an array of pre-defined images and sets the image of the actor to the next image of the series.~~