Chapter 1

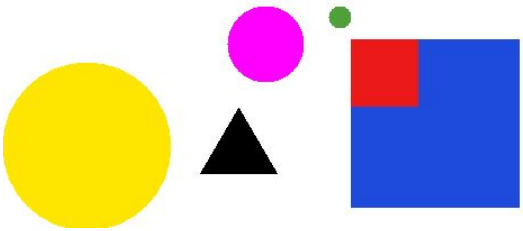
1.1: Created a circle and a Square

1.2: Calling moveDown twice moves the circle downwards each time. Calling makeInvisible twice makes the circle invisible after the first one and doesn't do anything the user could see the second time.

1.3: moveVertical moves an object instantly while slowMoveVertical moves it over time. Calling changeSize alters the size of the object. If we wanted to move an object horizontally to the left by 70 units we would call moveHorizontal and input (-70).

1.4: using changeColor I was able to make an object yellow, green and back to blue.
1.5: I was not able to make an object orange or purple.
1.6: without quotes an error message is received

1.7: Made some art



1.8: Calling moveLeft moved an object by -20 units

1.9: Step1 create all objects
- -square x2
- -triangle
- -circle

Step 2 change colors
- -make the circle yellow
- -make one square black

Step 3 change size and shape
- -make 1st square large and second one small
- -change the triangle to be wider than the big square

Step 4 moves shapes around
- - move small square over big one
- - move triangle over large square
- - move circle above to the right

1.10: some examples of the terminal view i tried

```
square1.moveDown();
triangle1.moveRight();
Square square3 = new Square();
square3.changeColor("yellow");
square3.slowMoveHorizontal(90);
```

There are many ways and orders to make either of those pictures

1.11: brought the code pod up
1.12: typed in

```
person person1 = new Person();
person1.makeVisible();
person1.moveRight();
```

1.13: created a picture invoked its draw method and set its color to black and white and then back to its default color
1.14: picture has a draw method that creates all the shapes at once with the correct parameters to make them the right size, shape, color, and position to make the picture.

1.15: pulled up the source code for the class picture

1.16: made the sun blue

```
sun.changeColor("blue");
```

1.17: added a second sun to the picture and made it visible

```
private Circle sun2;
sun2 = new Circle();

sun2.changeColor("yellow");
sun2.moveHorizontal(-175);
sun2.moveVertical(-20);
sun2.changeSize(90);
sun2.makeVisible();
```

1.18/1.19: added a sunset method that invokes slowMoveVertical on the sun until it leaves the screen

```
public void sunset()
{
 sun.slowMoveVertical(250);
}
```

1.20: added to sunset method and had a person appear after the sun lowers and the person aproaches the house.

```
public void sunset()
  {
   sun.slowMoveVertical(250);
   person1.makeVisible();
   person1.moveHorizontal(175);
   person1.slowMoveHorizontal(-145);

  }
```

1.21: created a student object
1.22: calling the getName method of student objects returns us the string that is holding the name for the student

1.23: created LabClass object and passed it an int for max number of students in the class
1.24: calling the numberOfStudents method returns us the int that is representing the maximum number of students the was entered when creating the object
1.25: added a student to the class using enrollStudent method
1.26: called printList method of the LabClass object and got a list of all students in the class

Lab class unknown
Instructor: unknown Room: unknown
Class list:
Mantis Toboggan, student ID: 5619, credits: 0
Number of students: 1

1.27: created a LabClass with all three students and added the appropriate credits.

Lab class unknown
Instructor: unknown Room: unknown
Class list:
Snow White, student ID: A00234, credits: 24
Lisa Simpson, student ID: C22044, credits: 56
Charlie Brown, student ID: A12003, credits: 6
Number of students: 3

1.28: LabClass object has 5 fields

Instructor
Room
Time and day
A list of students
Capacity

1.29: completed the missing fields for LabClass

Lab class Monday, 11am
Instructor: Dr. Mantis Toboggan Room: A202
Class list:
Snow White, student ID: A00234, credits: 24
Lisa Simpson, student ID: C22044, credits: 56
Charlie Brown, student ID: A12003, credits: 6
Number of students: 3

1.30: different basic data types differ in size and how many digits each data type can represent
1.31:

0	int
“Hello”	string
101	int
-1	int
True	boolean
“33”	string
3.1415	float

1.32: to add a new field called name to a circle object would be adding
Private String name to the circle class
1.33:Void send(string s)
1.34:int average (int p1, int p2)
1.35: The physical book would be an object and its class would be books examples would be other books
1.36: no a class is a blueprint for one specific object with a specific name