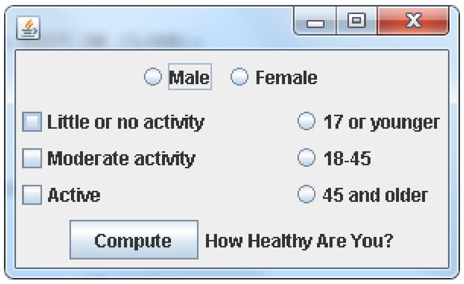
CSC 360 Midterm 2 answer key
Illegible answers may receive no credit. Show work for partial credit. Write on back of a page as needed.

1. Provide the code needed to create the following GUI. Only provide the code to build the JPanel that you might write in this inner class' constructor (you do not need to write the constructor's header or the { }). Do not bother with any other code. Do not define any listeners, just the code to create and properly position the GUI components. (15 points)

   NOTE: for convenience, abbreviate javax.swing components such as JCB for JCheckBox and JL for JLabel.



```
JButton c=new JButton(“Compute”);
JLabel lab=new JLabel(“How Healthy Are You?”);
JRadioButton male=new JRadioButton(“Male”);
JRadioButton female=new JRadioButton(“Female”);
JRadioButton age1=new JRadioButton(“17 or younger”);
JRadioButton age2=new JRadioButton(“18-45”);
JRadioButton age3=new JRadioButton(“46 or older”);
ButtonGroup g1=new ButtonGroup();
g1.add(male);
g1.add(female);
ButtonGroup g2=new ButtonGroup();
g2.add(age1);
g2.add(age2);
g2.add(age3);
JCheckBox b1=new JCheckBox(“Little or no activity”);
JCheckBox b2=new JCheckBox(“Moderate activity”);
JCheckBox b3=new JCheckBox(“Active”);
JPanel p1=new JPanel();
p1.add(male);
p1.add(female);
JPanel p2=new JPanel(new GridLayout(3,1));
p2.add(b1);
p2.add(b2);
p2.add(b3);
JPanel p3=new JPanel(new GridLayout(3,1));
p3.add(age1);
p3.add(age2);
p3.add(age3);
JPanel p4=new JPanel();
p4.add(c);
p4.add(lab);
JPanel p5=new JPanel(new BorderLayout());
p5.add(p1,BorderLayout.NORTH);
p5.add(p1,BorderLayout.WEST);
p5.add(p1,BorderLayout.EAST);
p5.add(p1,BorderLayout.SOUTH);
add(p5);
```

2. We have a Ball object called ball. The Ball class has methods to move( ) the Ball on the screen, getX( ), getY( ) to return the x and y coordinates, and bounce( ) to change the Ball's trajectory if it hits a wall. Our "screen" has borders of x=100, y=200, x=500, y=800. The Ball ball should move every time the Timer sends out an Event. Implement the actionPerformed method to move the ball while ensuring the ball to bounce off walls if in moving the Ball it hits a wall. (6 points)

```
public void actionPerformed(ActionEvent e)
{
      ball.move();
      if(ball.getX()<100||ball.getX()>500||ball.getY()<200||
                ball.getY()>800)
            ball.bounce();
      repaint();
}
```

3. You want to implement MouseListener in the current class.
   a. What instruction do you use to add this listener? (2 points)
      addMouseListener(this);

   b. What methods do you need to implement to implement MouseListener? (5 points)
      mousePressed, mouseClicked, mouseReleased, mouseExited, mouseEntered

   c. What type of Event do these methods receive? (2 points)
      MouseEvent

   d. In writing the methods listed in part b, which one of the following is true? (2 points)
      i. Every method must be listed but you can end the method header with ;
      [✓ ii. Every method must be implemented but the implementation can be { } (that is, empty)]
      iii. Every method must be implemented with instructions placed in { }
      iv. You do not need to list all of the methods but you must but you must list at least one (it can end with ;)
      v. You do not need to list all of the methods but you must but you must list at least one and it must have at least { }

4. We usually call `super.paintComponent(g);` in paintComponent. What effect does it have? Provide an example of why you would not use this in the paintComponent method. (5 points)

   It causes the Graphics object to be cleared. We would not do this if we wanted to continue to draw objects on top of previous ones such as in a paint program.

5. We want to define an abstract class called Employee. The Employee class will have Strings `name`, and `position`, a 1-arg constructor receiving a name (position will be unknown), a 2-arg constructor, accessors and mutators for both instance data, and an abstract method called `maximumSalary` (which receives no parameters and returns an int).

   a. Write this class. (10 points)

```
public abstract class Employee
{
        private String name, position;
        public Employee( ) {
                name=“unknown”;
                position=“unknown”;
        }

        public Employee(String n) {
                name=n;
                position=“unknown”;
        }

        public Employee(String n, String p) {
                name=n;
                position=p;
        }

        public String getName() { return name; }
        public String getPosition() { return position; }

        public void setName(String n) {name=n;}
        public void setPosition(String p) {position=p;}

        public abstract int maximumSalary( );
}
```

   b. Answer the following true/false questions (2 points each)

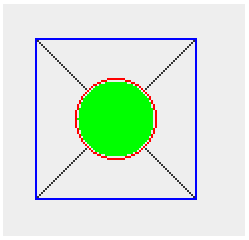
      i. [✓ True] or false: We can declare a variable of type Employee.

      ii. True or [✓ false]: We can instantiate a variable of type Employee.

      iii. True or [✓ false]: If we extend Employee, the extended class must implement maximumSalary.

      For iii, we must implement maximumSalary if the extended class is not abstract, but we could extend Employee into another abstract class which would not have to implement maximumSalary.

6. Implement the paintComponent method for a JPanel to draw the following image below to the right. Only write the paintComponent method. The rectangle is blue, the outline of the circle is red and the inner part of the circle is green. The lines are black. The rectangle is 80x80 starting at location 20,20, and the circle is 40x40. You will have to use a little math to figure out the remaining locations if needed. (10 points)



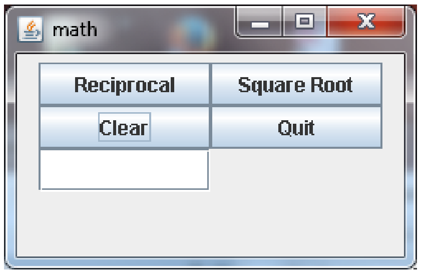
```
public void paintComponent(Graphics g)
{
   g.setColor(Color.blue);
   g.drawRect(20,20,80,80);
   g.setColor(Color.black);
   g.drawLine(20,20,80,80);
   g.drawLine(20,80,80,20);
   g.setColor(Color.red);
   g.drawOval(40,40,40,40);
   g.setColor(Color.green);
   g.fillOval(41,41,38,38);
 }
```

NOTE: there are only 2 lines here, not 4 (the circles are drawn over the lines)

7. What does setFocusable(true) do? When, why and to what object would you use this? (6 points)

It makes the JFrame or JPanel look for key entries. We would do this in main to the JFrame or JPanel if we were implementing KeyListener.

8. Given the GUI shown on the right with JButtons named `recip` (for reciprocal), `sqrt` (square root), `clear` and `quit`, a JLabel called `out`, and a JTextField called `in`, write the actionPerformed method to handle the 4 JButtons. Assume the input will always be an int value (not a double). Use Math.sqrt for square root. Output the word "Error" to `out` if either the input is 0 for Reciprocal or the input is negative for Square Root. (10 points)



```
public void actionPerformed(ActionEvent e) {
   int x=Integer.parseInt(in.getText());
   if(e.getSource()==recip)
         if(x==0) out.setText("Error");
         else out.setText(1.0/x);
   else if(e.getSource()==sqrt)
         if(x<0) out.setText("Error");
         else out.setText(Math.sqrt(x));
   else if(e.getSource()==clear) {
         in.setText("");
         out.setText("");
   }
   else System.exit(0);
}
```

9. We want to define a Person class which has instance data of `name` (String) and `age` (int). Person will have 3 constructors, a no-arg constructor which initializes name to "unknown" and age to 0, a 1-arg constructor receiving the Person's name, initializing age to 0, and a 2-arg constructor. We want to ensure that a Person does not have an illegal value for `age`. Legal values for `age` are 0 to 120.

   a. Implement an Exception class called IllegalAge which consists only of the proper constructor(s). (6 points)

```
public class IllegalAgeException extends Exception
{
      public IllegalAgeException( ) {
            super("Age not legal for a human");
      }
      public IllegalAgeException(String s) {
            super(s);
      }
}
```

   b. Implement the Person's `setAge` mutator which needs to ensure a proper age or else it will have to throw an appropriate Exception. Implement this method. NOTE: do not handle an Exception in a try-catch block in this method! (6 points)

```
      public void setAge(int a) throws IllegalAgeException {
            if(a<0||a>120) throw new IllegalAgeException(a +
                  " is not a legal human age!");
            else age=a;
      }
```

   c. For Person, the class header is `public class Person implements Comparable`. Implement the compareTo method. Person's will be compared by `age` first and if the two Person's ages are the same, then compare `name`. Remember there are accessors for Person's `age` and `name`.(6 points)

```
      public int compareTo(Object o)
      {
            Person p=(Person)o;
            if(age>p.getAge()) return 1;
            else if(age<p.getAge()) return -1;
            else return name.compareTo(p.getName());
      }
```

   d. We want to implement a class called People which will create an array of Person objects. Our main method has the following declarations.

```
      Person[] people=new Person[1000];
      int numberOfPeople=0;
      Scanner in=new Scanner(System.in);
```

   We want to input new `Person` objects into the `people` array. This code is placed into a try-catch block. There are at least three catches that we want to try to catch. What are they? (3 points)

```
      IllegalAgeException
      ArrayIndexOutOfBoundsException
      InputMismatchException
```