SE1021 Exam 2 Name:

You may use a note-sheet for this exam. But all answers should be your own, not from slides or text. Review all questions before you get started. The exam is printed single-sided. Write your name on each page. You may do this before the exam if you don't peek at the questions.

When returning your exam, place your note-sheet on top.

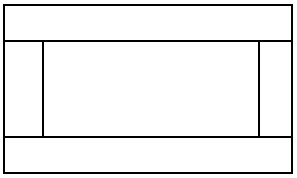
Name:_____

1. (4 pts) Multiple Choice
    a. Would you extend or implement JFrame?
        i. extend
        ii. implement
    b. Would you extend or implement ActionListener?
        i. extend
        ii. implement
    c. Would you extend or implement Exception?
        i. extend
        ii. implement
    d. Which provides a specific response to an event?
        i. the source object
        ii. the listener object
2. (6 pts.) Multiple Choice
    a. Select the option which completes the following code-snippet so that when the button is clicked, the `listener` can handle the method.

        JButton b = new JButton("b");

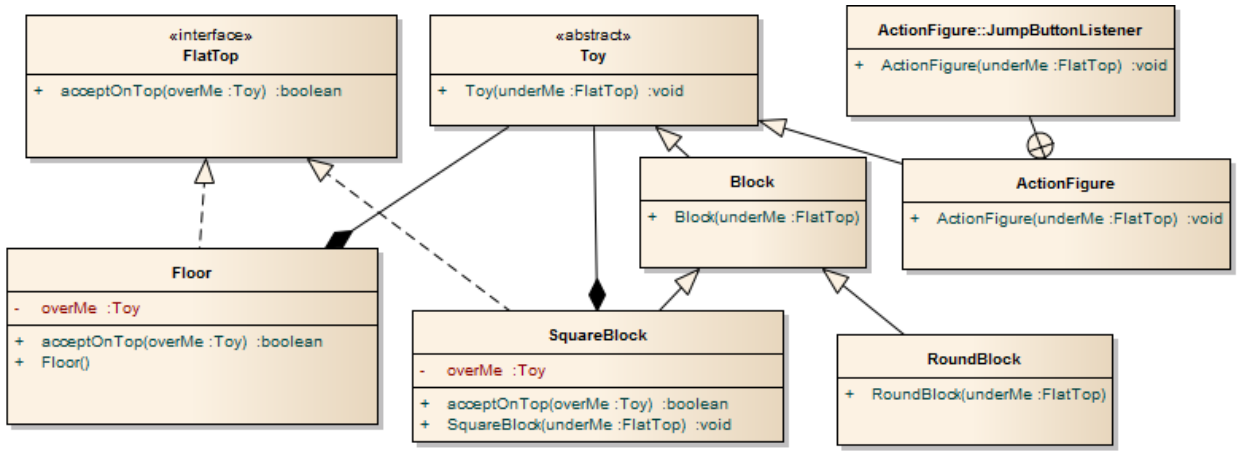
        _____

        i. addEventListener(listener);
        ii. addActionListener(listener);
        iii. b.addEventListener(listener);
        iv. b.addActionListener(listener);
    b. Which one of the following will compile?
        i. public abstract class A { public abstract void fun(); }
        ii. public abstract class A { public abstract void fun() { } }
        iii. public class A { public abstract void fun(); }
        iv. public class A { public abstract void fun() { } }
    c. Which flow manager would be best for designing this layout:



        i. FlowLayout
        ii. BorderLayout
        iii. BoxLayout
        iv. GridLayout

Name:_____



3. (10 pts.) See the UML diagram above. Select one.
    a. The relationship between Toy and Floor is
        i. Composition
        ii. Aggregation
        iii. Inner-class
        iv. Inheritance
        v. Implementation
    b. The relationship between JumpButtonListener and ActionFigure is
        i. Composition
        ii. Aggregation
        iii. Inner-class
        iv. Inheritance
        v. Implementation
    c. Which one of the following statements is valid?
        i. Toy t = new Block(new Floor());
        ii. Toy t = new Toy();
        iii. Block b = new Block();
        iv. Block b = new Toy(new Floor());
    d. Which one of the following statements is valid?
        i. FlatTop t = new FlatTop(new Floor());
        ii. FlatTop t = new SquareBlock();
        iii. SquareBlock f = new SquareBlock(new Floor());
        iv. SquareBlock f = new FlatTop();
    e. Which one of the following method-calls is valid? Assume `b` and `b2` are both `SquareBlocks` and `f` is a `Floor`.
        i. b.acceptOnTop(new FlatTop())
        ii. b.acceptOnTop(b2)
        iii. b.acceptOnTop(f)
        iv. b.acceptOnTop(new Toy(f))

Name:_____

4. [20 pts.] Answer clearly & concisely
   a. (3 pts.) Consider this code:

```
public abstract class Shape {
    public abstract void draw();
}

// In another file…
public ExamFun {
    public abstract void drawShapes(Shape shape1, Shape shape2) {
        shape1.draw();
        shape2.draw();
    }
}
```

   If used correctly with other code, this example will both compile and run. How is this possible, since Shape does not supply a body for the draw method?

   b. (3pts.) Name two Swing classes that act as event sources.

   c. (3 pts.) While implementing an actionPerformed method, we could use `e.getSource().equals(r1)` instead of `e.getSource() == r1` where `r1` is a reference to one of the event sources in our program. Why might the `.equals` solution return `true` even though == returns false?

   d. (3 pts.) What happens if an unchecked exception is thrown but never caught?

   e. (4 pts.) What are two differences between an anonymous inner class and a regular inner class?

   f. (4 pts.) On which line and in which method did the error occur? What happened?

```
Exception in thread "AWT-EventQueue-0" java.lang.ArithmeticException: / by zero
      at shapes.Ellipse.draw(Ellipse.java:36)
      at shapes.ShapeManager.draw(ShapeManager.java:113)
      at ui.DrawingProgramUI.paint(DrawingProgramUI.java:133)
```

Name:_____

5. (15 pts) The following code compiles and runs, but the buttons don't do anything when you click on them. Fix the problem so the program responds to both buttons.

```
// You can assume these are correct
import ...

public class FunGui extends JFrame {

    JButton b1 = new JButton("press me");
    JButton b2 = new JButton("press me");
    JLabel label = new JLabel("The button pressed will be displayed here.");

    public FunGui() {
        setTitle("Fun GUI");
        setLayout(new FlowLayout());



        b1 = new JButton("press me");
        b2 = new JButton("press me");
        label = new JLabel("The button pressed will be displayed here.");



        add(b1);
        add(b2);
        add(label);



        pack();
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        if(e.getSource() == b1) {
            label.setText("Button 1 pressed");
        } else if(e.getSource() == b2) {
            label.setText("Button 2 pressed");
        }
    }

    public static void main(String[] args) {
        new FunGui();
    }
}
```

Name:_____

6. [15 pts.] Consider the code below:

```
public static void main(String[] args) {
    String str = JOptionPane.showInputDialog(null,"Enter something");
    try {
        int i = Integer.parseInt(str);
        int ratio = 10/i;
        System.out.println("ratio: "+ratio);
    } catch (ArithmeticException e) {
         System.out.println("ouch");
    } catch (NumberFormatException e) {
         System.out.println("eek");
    }
    System.out.println("done");
}
```

   a. (5 pts.) How can the user get the program to print "`ratio: 5`?"

   b. (5 pts.) How can the user get the program to print "`eek`?"

   c. (5 pts.) What can the user enter, if anything, that would cause the program to not print "`done`?"

Name:_____

7. (10 pts.) What does the following code print?

```
public class A {
    public void m() {
        System.out.print("1 ");
    }

    public void m(int argument) {
        System.out.print("2 ");
        System.out.print(argument+ " ");
    }
}

public class B extends A {
    public void m() {
        this.m(101);
        System.out.print("3 ");
    }

    public void m(int argument) {
        super.m();
        System.out.print("4 ");
        System.out.print(argument+ " ");
    }

    public static void main(String[] ignored) {
        B b = new B();
        b.m();
    }
}
```

8. (10 pts.) Create a customized class, `ExamException`, that can be thrown by the following command:

```
   throw new ExamException();
```

Users should not have to explicitly handle this exception.