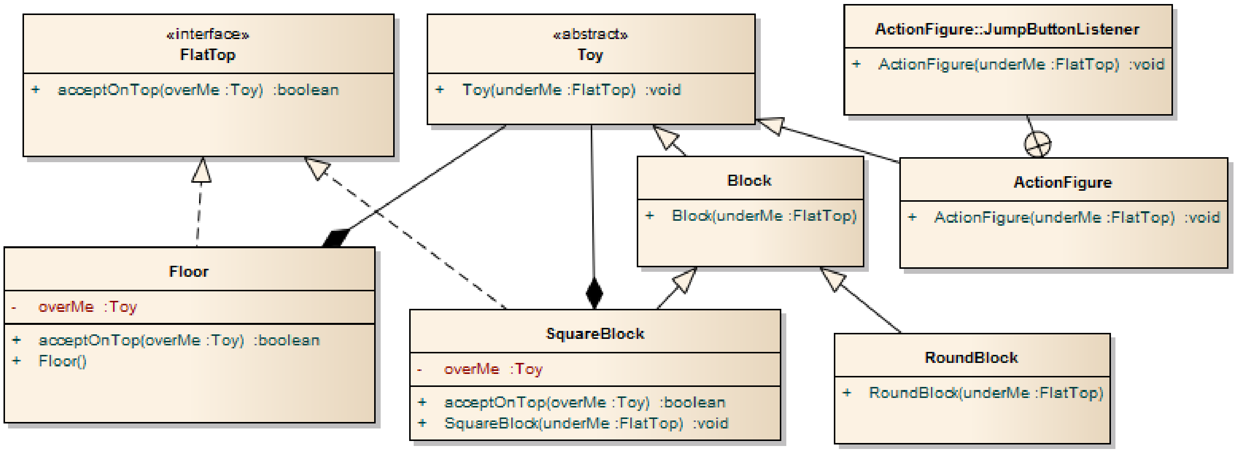
Name:______________________________

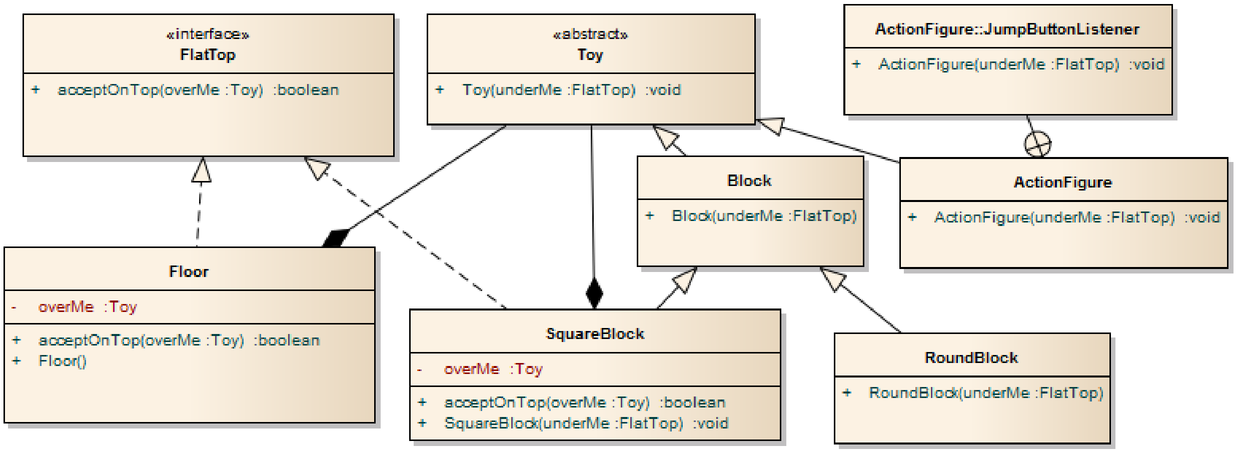
SE1021 HW Week 8 (4th) Name:

1. Why do options ii-iv not 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() { } }


«interface»
FlatTop
+ acceptOnTop(overMe :Toy) :boolean
«abstract»
Toy
+ Toy(underMe :FlatTop) :void
ActionFigure::JumpButtonListener
+ ActionFigure(underMe :FlatTop) :void
Block
+ Block(underMe :FlatTop)
ActionFigure
+ ActionFigure(underMe :FlatTop) :void
Floor
- overMe :Toy
+ acceptOnTop(overMe :Toy) :boolean
+ Floor()
SquareBlock
- overMe :Toy
+ acceptOnTop(overMe :Toy) :boolean
+ SquareBlock(underMe :FlatTop) :void
RoundBlock
+ RoundBlock(underMe :FlatTop)

2. See the UML diagram above. Select one
    - a. The relationship between Toy and SquareBlock is
        - i. Composition
        - ii. Aggregation
        - iii. Inner-class
        - iv. Inheritance
        - v. Implementation
    - b. The relationship between FlatTop and SquareBlock is
        - i. Composition
        - ii. Aggregation
        - iii. Inner-class
        - iv. Inheritance
        - v. Implementation
    - c. The relationship between Block and SquareBlock is
        - i. Composition
        - ii. Aggregation
        - iii. Inner-class
        - iv. Inheritance
        - v. Implementation

Name:____________________________________


«interface»
FlatTop
+ acceptOnTop(overMe :Toy) :boolean
«abstract»
Toy
+ Toy(underMe :FlatTop) :void
ActionFigure::JumpButtonListener
+ ActionFigure(underMe :FlatTop) :void
Block
+ Block(underMe :FlatTop)
ActionFigure
+ ActionFigure(underMe :FlatTop) :void
Floor
- overMe :Toy
+ acceptOnTop(overMe :Toy) :boolean
+ Floor()
SquareBlock
- overMe :Toy
+ acceptOnTop(overMe :Toy) :boolean
+ SquareBlock(underMe :FlatTop) :void
RoundBlock
+ RoundBlock(underMe :FlatTop)

3. (See above.) Select the correct option and explain why the others are wrong.

   a. Which one of the following statements is valid?

      i. FlatTop t = new FlatTop();
      ii. FlatTop t = new SquareBlock(new Floor());
      iii. SquareBlock f = new SquareBlock();
      iv. SquareBlock f = new FlatTop(new Floor());

      Explanation:

   b. Which one of the following statements is valid?

      i. Toy t = new Block();
      ii. Toy t = new Toy(new Floor());
      iii. ActionFigure b = new Block(new Floor());
      iv. ActionFigure b = new Toy();

      Explanation:

   c. Which one of the following method-calls is valid? Assume `b` is a `SquareBlock`, b2 is a RoundBlock, and `f` is a `Floor`.

      i. b2.acceptOnTop(b)
      ii. f.acceptOnTop(b2)
      iii. b2.acceptOnTop(new Floor())
      iv. f.acceptOnTop(new FlatTop())

      Explanation:

4. Answer clearly & concisely
   a. 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. The `.equals` solution might return `true` if someone overrides the .equals method. If you are trying to find the source of an event, which solution is preferable?

   b. When writing your own exception, how do you make it unchecked?

5. Create an anonymous inner class implementing "Runnable". This interface has a single method: void run(); Make a reference to an object of your inner class called `run`. Make sure to use the right type for the reference.

Name:____

6. (10 pts.) What does the following code print? Each class is declared in its own file with the proper imports.

```
public abstract class C {
    public C() {
        System.out.print("C1 ");
    }

    abstract void m();

    abstract void m(int argument);
}

public class A extends C {
    public void m() {
        System.out.print("A1 ");
    }

    public void m(int argument) {
        System.out.print("A2 ");
        System.out.print("A" + argument+ " ");
    }
}

public class B extends A {
    public void m() {
        super.m(101);
        System.out.print("B1 ");
    }

    public void m(int argument) {
        super.m();
        System.out.print("B2 ");
        System.out.print("B"+argument+ " ");
    }

    public static void main(String[] ignored) {
        C c = new B(); // Does this print anything?
        c.m();
        c.m(102);
    }
}
```