Registration Number: ____________________

In-class Test 2

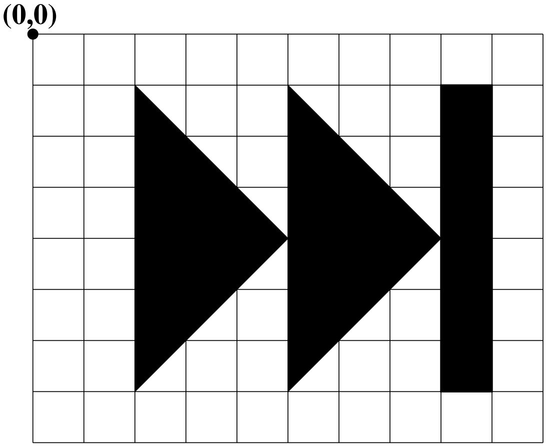
MSc/ICY Software Workshop

Assessed In-class Test: 10% of the continuous assessment mark.

Submission: Friday, 4 December 2015, 9:50 hours
No late submission

Usual examination conditions apply. You may not use any material during this in-class test.

Exercise 1: (Basic, 30%)



Assume that the grid shows a distance of 10 pixels on a panel. Write a method `public void paintComponent(Graphics g)` that draws the fast-forward symbol as displayed to the left on the panel (just the symbol, not the grid lines.)

```
public void paintComponent(Graphics g){
   super.paintComponent(g);



}
```

Exercise 2: **(Medium, 40%)** Assume that in a cinema context we have a Ticket class with field variables private String screen, private int price. Furthermore there is a constructor public Ticket(String screen, int price), and getters and setters for the two field variables. Also we have

```
public String toString(){
  return "Screen: " + getScreen() +
         ". Total Price: " + getPrice();
}
```

For 3D films we want to have a Ticket3D subclass that has the additional field variable private boolean needGlasses. If customers need glasses they are charged an extra fee stored in a static variable public static final int glassesFee = 2.

Write a Ticket3D class with an appropriate constructor and use overriding so that the toString() method prints the information in form of: "Screen: 3. Total Price: 9" (for a film to which the entrance fee is 7 and the customer needs to pay 2 for glasses).

Exercise 3: (Advanced, 30%)

Assume the following five separate Java files. `Main.java` contains a main method with four print statements. Write in the lower right box what will be printed in each case.

```java
public class A implements C {
    private int i;
    public  int j;
    public  int k;

    public A(int i, int j, int k) {
        this.i = i;
        this.j = j;
    }

    public int f(int k) {
        return this.i + this.j +
               this.k;
    }
}
```

```java
public class B extends A {
    public int k = 5;

    public B(int i, int j, int k) {
        super(i,k,-1);
        k = i + k;
    }

    @Override
    public int f(int k) {
        return super.f(k) + k;
    }
}
```

```java
public interface C {
    public static final int i = 6;
    public int f(int k);
}
```

```java
public class D extends B implements C {
    public D(int i, int j, int m) {
        super(i,j,m);
    }

    public int f(int k) {
        return super.f(k) +
               this.j * this.k;
    }
}
```

```java
public class Main{
    public static void
      main(String[] args) {
        A a = new A(1,2,3);
        System.out.println(a.f(4));

        B b = new B(1,2,3);
        System.out.println(b.f(4));

        C c = new B(1,2,3);
        System.out.println(c.f(4));

        D d = new D(1,2,3);
        System.out.println(d.f(4));
    }
}
```

____

____

____

____