Each question worth 5 points for total of 100

1. What is wrong in the following code?

```java
class Test {
    public static void main(String[] args) {
        A a = new A();
        a.print();
    }
}

class A {
    String s;
    A(String s) {
        this.s = s;
    }
    public void print() {
        System.out.print(s);
    }
}
```

2. (Geometry: The MyRectangle2D class) Define the MyRectangle2D class that contains:

- Two **double** data fields named `x` and `y` that specify the center of the rectangle with **get and set** methods. (Assume that the rectangle sides are parallel to `x`- or `y`- axes.)
- The **double** data fields `width` and `height` with **get and set** methods.
- A no-arg constructor that creates a default rectangle with `(0, 0)` for `(x, y)` and 1 for both `width` and `height`.
- A constructor that creates a rectangle with the specified `x`, `y`, `width`, and `height`.
- A method `getArea()` that returns the area of the rectangle.
- A method `getPerimeter()` that returns the perimeter of the rectangle.
- A method `contains(double x, double y)` that returns `true` if the specified point `(x, y)` is inside this rectangle. See Figure 1(a).
- A method `contains(MyRectangle2D r)` that returns `true` if the specified rectangle is inside this rectangle. See Figure 1(b).
- A method `overlaps(MyRectangle2D r)` that returns `true` if the specified rectangle overlaps with this rectangle. See Figure 1(c).
(a) A point is inside the rectangle. (b) A rectangle is inside another rectangle. (c) A rectangle overlaps another rectangle.

Draw the UML diagram for the class.

3. Analyze the following code:

```java
public class Test {
    private int t;

    public static void main(String[] args) {
        int x;
        System.out.println(t);
    }
}
```

A. The program compiles and runs fine.
B. The variable `t` is private and therefore cannot be accessed in the main method.
C. The variable `x` is not initialized and therefore causes errors.
D. `t` is non-static and it cannot be referenced in a static context in the main method.
E. The variable `t` is not initialized and therefore causes errors.

4. What is the output of the following program?

```java
public class Foo {
    private static int i = 0;
    private static int j = 0;

    public static void main(String[] args) {
        int i = 2;
        int k = 3;
        {
            int j = 3;
            System.out.println("i + j is " + i + j);
        }
        k = i + j;
        System.out.println("k is " + k);
        System.out.println("j is " + j);
    }
}
```
5. The `java.util.Date` class implements `java.lang.Cloneable` and overrides the `equals` method to return true if two objects have the same date and time. Show the output of the following code.

```java
import java.util.*;
public class Test extends Object {
    public static void main(String[] args) {
        Date d1 = new Date();
        Date d2 = new Date(349324);
        Date d3 = d1;
        System.out.println("(1) " + (d1 == d2));
        System.out.println("(2) " + (d1 == d3));
        System.out.println("(3) " + d1.equals(d2));
        System.out.println("(4) " + d1.equals(d3));
    }
}
```

6. Show the output of running the class Test in the following code:

```java
interface A {
    void print();
}
class C {}
class B extends C implements A {
    public void print() {
    }
}
public class Test {
    public static void main(String[] args) {
        B b = new B();
        if (b instanceof A)
            System.out.println("b is an instance of A");
        if (b instanceof C)
            System.out.println("b is an instance of C");
    }
}
```

- b. b is an instance of A.
- c. b is an instance of C.
- d. b is an instance of A followed by b is an instance of C.
7. What is the output of running class C?

class A {
    public A() {
        System.out.println("The default constructor of A is invoked");
    }
}

class B extends A {
    public B(String s) {
        System.out.println(s);
    }
}

public class C {
    public static void main(String[] args) {
        B b = new B("The constructor of B is invoked");
    }
}

a. none  
b. "The constructor of B is invoked"  
c. "The default constructor of A is invoked" "The constructor of B is invoked"  
d. "The default constructor of A is invoked"

8. Analyze the following code.

class Test {
    public static void main(String[] args) {
        Object x = new Integer(2);
        System.out.println(x.toString());
    }
}

a. The program has syntax errors because an Integer object is assigned to x.  
b. When x.toString() is invoked, the toString() method in the Object class is used.  
c. When x.toString() is invoked, the toString() method in the Integer class is used.  
d. None of the above.

9. Describe the role of the this keyword. What is wrong in the following code?

```java
public class C {
  int p;

  public C() {
    System.out.println("C’s no-arg constructor invoked");
    this();
  }

  public C(int p) {
    p = p;
  }

  public void setP(int p) {
    p = p;
  }
}
```
10. You can derive a new class from an existing class.
   a. true
   b. false

11. The keyword this can be used inside a constructor to invoke another constructor of the same class.
   a. true
   b. false


15. The keyword instanceof is used to determine
   a. the name of a class
   b. the class of an object
   c. the superclass of an object
   d. none of the above

16. The equals method in the Object class compares
   a. content of two objects
   b. reference variables of two objects
   c. classes of two objects

17. Use the ___________ modifier to prevent extending or overriding classes or methods.

18. Every class in Java is descended from the java.lang.Object class.
   a. true
   b. false

19. Explicit casting is always needed when
   a. casting subclass to superclass
   b. casting superclass to subclass
   c. overriding a method
   d. none of the above

20. An instance method can be overridden only if it is accessible.
   a. true
   b. false
Practice Exam 2 Key

1. No no-arg constructor
2. See next page with UML
3. D
4. i + j is 23
   k is 2
   j is 0
5. (1) false
   (2) true
   (3) false
   (4) true
6. D
7. C
8. C
9. call to this must be first statement in constructor
   ```java
   public C() {
       this(0);
       System.out.println("C's no-arg constructor invoked");
       // this(0);
   }
   ```
10. A
11. A
12. Ability to create a variable, a function, or an object that has more than one form.
13. Restricting access to an objects components.
14. Deriving classes from other classes creating subclasses.
15. B
16. B
17. final
18. A
19. B
20. A
UML Diagram for problem 2.

<table>
<thead>
<tr>
<th>MyRectangle2D</th>
</tr>
</thead>
<tbody>
<tr>
<td>-x: double</td>
</tr>
<tr>
<td>-y: double</td>
</tr>
<tr>
<td>-width: double</td>
</tr>
<tr>
<td>-height: double</td>
</tr>
<tr>
<td>+MyRectangle2D()</td>
</tr>
<tr>
<td>+MyRectangle2D(x: double, y: double, height: double, width: double)</td>
</tr>
<tr>
<td>+getX(): double</td>
</tr>
<tr>
<td>+getY(): double</td>
</tr>
<tr>
<td>+getHeight(): double</td>
</tr>
<tr>
<td>+getWidth(): double</td>
</tr>
<tr>
<td>+setX(x: double): void</td>
</tr>
<tr>
<td>+setY(y: double): void</td>
</tr>
<tr>
<td>+setHeight(height: double): void</td>
</tr>
<tr>
<td>+setWidth(width: double): void</td>
</tr>
<tr>
<td>+getArea(): double</td>
</tr>
<tr>
<td>+getPerimeter(): double</td>
</tr>
<tr>
<td>+contains(x: double, y: double): boolean</td>
</tr>
<tr>
<td>+contains(r: MyRectangle2D): boolean</td>
</tr>
<tr>
<td>+overlaps(r: MyRectangle2D): boolean</td>
</tr>
</tbody>
</table>