

Sample Placement Exams for Placement out of CSCI-UA.101:

Topics Covered by the Java Placement Exam

The Java version of the placement exam covers the following topics:

- Control structures: `if` statements, `switch` statement, `for`, `while` and `do...while` loops.
- Simple and multi-dimensional array
- Object oriented design
 - class design and use
 - encapsulation
 - reference and object manipulation
 - static vs. instance data fields and methods
 - arrays of objects
 - inheritance and polymorphism
 - abstract classes
- Exception handling
- File I/O
- Interfaces

Students should also be familiar with the basic Java classes like `String`, `Scanner`, `ArrayList<E>`, `File`, wrapper classes (`Integer`, `Float`, `Character`) and the `Comparable<E>` interface.

Reference: any book on Java programming and object oriented design.

Sample Questions for the Java Placement Exam

1. Show the output of the following code:

```
public class Test{
    public static void main ( String [] args ) {
        int [] a = { 1, 2 };
        swap ( a[0], a[1]);
        System.out.println(a[0] + " " + a[1] );
    }

    public static void swap ( int n1, int n2 ) {
        int temp = n1;
        n1 = n2;
        n2 = temp;
    }
}
```

2. What is the output of running the class C.

```
public class C {
    public static void main(String[] args) {
        Object o1 = new A();
        Object o2 = new B();
        System.out.print(o1);
        System.out.print(o2);
    }
}

class A extends B {
    public String toString() {
        return "A";
    }
}
```

```

class B {
    public String toString() {
        return "B";
    }
}

```

3. Write a Java class `Point` that represents (x,y) point in a plane. The class should implement `Comparable<Point>` interface. The points should be compared based on their distance from the origin (point (0,0)). The distance from the origin can be computed using $\text{distance} = \sqrt{x^2 + y^2}$.

Your class should implement all methods needed for the following code to compile and run successfully:

```

Random r = new Random();
Point [] myPoints = new Point[10];
for (int i = 0; i < myPoints.length; i++)
    myPoints[i] = new Point(r.nextDouble(), r.nextDouble() );
Arrays.sort(myPoints);

```

You do not need to provide any additional methods.

4. Write a method that, given an array of `Circle` objects, sorts the circles from smallest to largest. Assume that the `Circle` class is defined as follows:

```

public class Circle {
    private float radius;
    public Circle ( float r ) {
        if ( r >= 0 ) radius = r;
        else radius = 1;
    }
    public float getRadius() {
        return radius;
    }
    public int compareTo( Circle c ) {
        if (radius == c.radius ) return 0;
        else if (radius < c.radius ) return -1;
        else return 1;
    }
}

```

You should use the following signature for the method:

```

void sortCircles ( Circle [] circles )

```

You can use any sorting algorithm we discussed.