No. of Pages	**2**
No. of Questions	**3**

Department of Computer Science and Engineering
FINAL EXAMINATION Fall 2020

CSE310: Object Oriented Programming
Total Marks: 100 Time Allowed: 1 Hour

- Answer all **Three (3)** questions
- Figure in bracket [] next to each question indicates marks for that question
- **Precaution**: All descriptions should be as brief as possible and to the point. Please make them at most 2 or 3 sentences. GOOD LUCK ☺

Name: _____
ID: _____
Section: 1

Question 1 [20 Points]

A. When is the java toString() method automatically runs? Give simple example.

B. What are Exceptions? Why Use Exceptions? What is difference between error and exception? When the finally block is executed?

Question 2 [40 Points]

Answer **Four** questions below:

A. Why do we need synchronization? Describe with the help of a scenario.
B. A class is written as *protected* modifier. Who has the privileges to access the class?
C. What will happen to a class if it has the only constructor as private?
D. Describe Daemon Thread. What is purpose of this thread type?
E. A extends B, B extends C. When will this work and why?
[code: *A a = New C(); a.method();]*
F. Define method overloading and overriding?

Question 3 [40 Points]

Given classes below, you have to write the missing classes to make this program work without error and shows the following output. You **do not change** given classes. You **do not** have to **rewrite** the given classes.

```
public class ShapeDemo {
   public static void main(String[] args) {
      Shape c = ShapeFactory.getInstance("c", 5.0);
      Shape r = ShapeFactory.getInstance("r", 3, 4);
      Shape s = ShapeFactory.getInstance("s", 6);

      ShapePrinter.print(c);
      ShapePrinter.print(r);
      ShapePrinter.print(s);
   }
}
```

```
public class ShapePrinter {
   public static void print(Shape shape) {
      System.out.println("Shape: " + shape);
      System.out.println("AREA: " + shape.getArea());
      System.out.println("PERIMETER: " + shape.getPerimeter());
      System.out.println();
   }
}
```

```
public interface Shape {
   double getArea();
   double getPerimeter();
}
```

Output:

```
Shape: Circle
AREA: 78.53981633974483
PERIMETER: 31.41592653589793

Shape: Rectangle
AREA: 12.0
PERIMETER: 14.0

Shape: Square
AREA: 36.0
PERIMETER: 24.0
```