CST8288 (15F) Midterm Test: Student Name _____

Test Version A

Instructions (Read These):

- **Starting the test before permission has been given will result in an immediate zero. Everyone starts at the same time once everyone is seated.**
- **Turn mobile phones / mobile devices off**
- **Remove all papers, books, laptops etc. from the desks**
- **Using any electronic device – laptop, mobile phone, PDA, calculator during the test will result in immediate zero**
- **Strictly no talking for any reason during the test or you get zero, unless to ask the professor a question If you complete your test early, do not talk to friends who are still taking the test.**
- **Write your name at the top of the test document on page 1 and at the end for part 2, as well as on the Multiple Choice Answer Sheet.**
- **Assume code samples shown will run / compile as part of a larger code listing, unless there is an option provided "does not compile" or similar for a possible answer**
- **All questions are within the context of Java, and computer programming**
- **If you see what you think may be a mistake raise your hand and ask quietly when I reach you**
- **Please do not leave the room during the first 30 minutes of the test.**
- **All test materials must be returned at the end of the test**

Additional Notes:
(**No Calculators allowed)**

The test is in two parts:

Part 1 is multiple-choice questions

- Place your answers for Part 1 onto the provided answer sheet, only the answer sheet will be used for grading Part 1
- Note: Make sure your handwriting is readable; I suggest using upper case letters ensuring that D looks different from B.

Part 2 is short answer

- Place your answers for Part 2 into the spaces provided. If you need more room clearly note that your answer is continued on extra paper, and on the extra paper clearly indicate what question you are continuing.

- There is an extra blank page at the end of the test you can detach to use as scrap paper, remember to return all sheets at the end of the test, including scrap paper.

Ensure that your name is on all parts of the test:

- Multiple Choice answer sheet
- The test (this document)
- Any extra sheets you detach

Total Test Time will be 1 hour 50 minutes

Part 1: Multiple Choice Questions 60 points [2 points each]

1. Which of the following is a benefit of the OOP approach to programming?
   a) Code maintainability
   b) Code reusability
   c) Code reliability
   d) All of the above

2. In an Object Oriented programming language data and related functions are combined into a single unit called a(n) _____.
   a) Token
   b) Object
   c) Procedure

3. Objects communicate by sending messages between each other, this is implemented using _____.
   a) Variables
   b) Methods
   c) Reflection

4. Which of the following is not one of Java's access modifiers?
   a) public
   b) protected
   c) internal

5. Which sentence describes Principle of Least Privilege best?
   Each component should have:
   a) full rights and privileges so it can do its task
   b) minimal rights and privileges or full rights and privileges so it can do its task
   c) minimal rights and privileges to accomplish its task, but no additional rights or privileges

6. Which of the items below is often described as a blueprint?
   a) Class
   b) Object
   c) Enum

7. Given the classes below what prints from the program when it is run?

```
public class Counting{
    public static int count;
    public Counting(){ count++; }
}
```

```
public class CountingDemo{
    public static void main(String[] args){
        System.out.print( (new Counting() ).count );
        System.out.print( (new Counting() ).count );
}}
```

   a) 00
   b) 11
   c) 12

8. Which of the following terms is used to categorize design patterns related to object instantiation?
   a) Behavioural
   b) Structural
   c) Creational

9. Which design pattern from the list below is concerned with separating construction of a complex object from it's implementation?
   a) Singleton
   b) Builder
   c) Delegate

10. The code sample below is for a Singleton design pattern. What one thing is wrong?

```
public class SingleThing {
    private static final SingleThing thing = new SingleThing();
    public SingleThing(){} // private constructor
    public static SingleThing instance(){return thing;}
}
```

   a) The method named instance() should not be public
   b) The constructor was marked public, it needs to be private
   c) You cannot assign a new SingleThing reference to variable thing because it was marked final

11. This code, taken from Assignment 2 suggests a(n) _____ design pattern

```
public void setBandwidth(double bandwidth){
    model.setBandwidth(bandwidth);
}
```

   a) Singleton
   b) Delegate
   c) Builder

12. The code sample below is taken from delegate design pattern demonstration, item AccountBehaviour is a(n) _____.

```
public class Account {
    private double balance;
    private AccountBehaviour accountBehaviour;
```

   a) Superclass
   b) Interface
   c) Enum

13. When using JUnit 4 what annotation is used to mark a test method?
   a) @Test
   b) @CheckCode
   c) @Testing

14. Which one of the JUnit methods below is best used for comparing two double values?
   a) assertEquals(String message, double expected, double actual)
   b) assertEquals(String message, double expected, double actual, double delta)
   c) assertTrue(String message, (expected == actual))

15. Which of the following is not one of the typical workflow steps when creating a JUnit test?
    a) Check results for the one task tested using an appropriate assert method.
    b) Prepare objects and variables (use meaningful variable names).
    c) Perform many tasks to be tested all at once inside one test method.

16. Which of the classes below represent an immutable (non-changeable) sequence of characters in a Java program?
    a) String
    b) StringBuilder
    c) StringBuffer

17. Which one of the methods of class StringBuilder is used to concatenate text into the StringBuilder?
    a) builder.append("text");
    b) builder.toString("text");
    c) builder.add("text");

18. What is wrong with the Java class declaration below, with regard to coding conventions?

    public class tuna_fish{}

    a) Class identifiers in Java should start with an upper case letter.
    b) Class identifiers in Java should use mixed case instead of underscores.
    c) Both of the above

19. Given the recursive example below what is printed to the console?

```
public class B{
    public static void main(String[ ] args){
        recurse(2);
    }
    public static void recurse(int n){
        if (n >= 0) {
            System.out.print(n + " ");
            recurse(n - 1);
            System.out.print(n + " ");
        }
    }
}
```

    a) 3 2 1 1 2 3
    b) 2 1 0 0 1 2
    c) 0 1 2 2 0 1

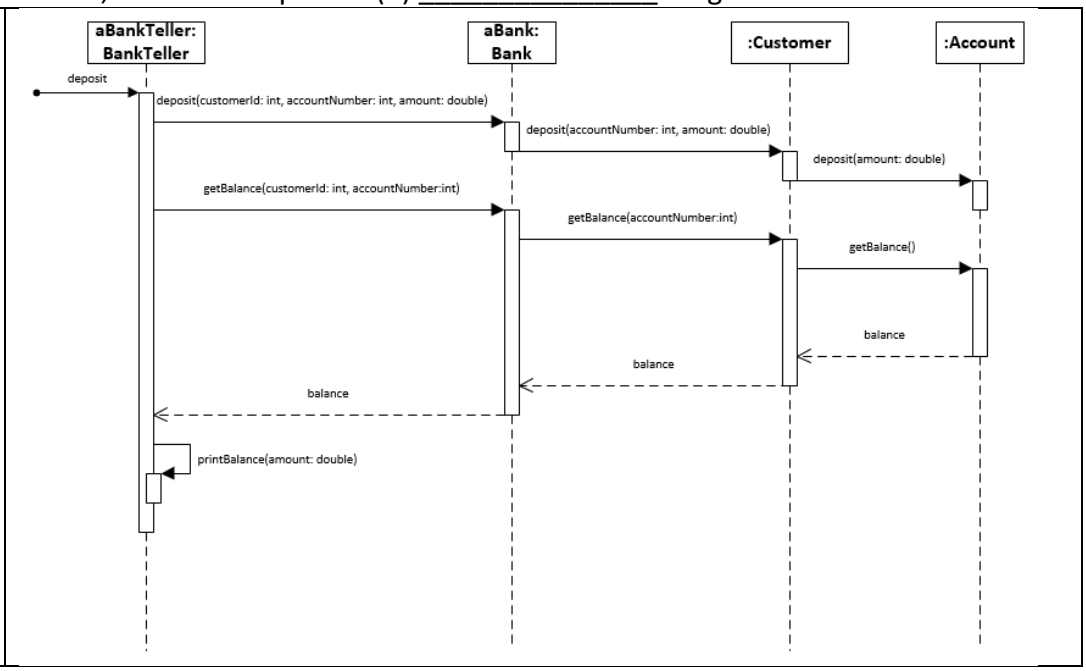
20. Which of the following lines below list Big-O in order from best to worse?
    a) O(1), O(log n), O(n), O(n log n), $O(n^2)$
    b) O(n), O(n log n), O(1), O(log n), $O(n^2)$
    c) $O(n^2)$, O(n log n), O(n), O(log n), O(1)

21. Which Abstract Data Type would you use for simulating people waiting in a line?
    a) Stack
    b) !n7t6d
    c) Vector

22. Which of the following array sorting algorithms has Big-O(n log n)?
    a) Selection Sort
    b) Bubble Sort
    c) Merge Sort

23. Examine the figure below, it is an example of a(n) _____ Diagram.

a) UML Sequence
b) UML Class
c) Use Case



24. The following code is an example of the _____ design pattern.

```
Comparator comp = new ComparatorAdapter(){
    @Override
    public boolean compare(double a, double b){
        return (a < b);
    }
};
```

    a) Singleton
    b) Builder
    c) Adapter

25. Which of the following swing GUI components allows a user to enter text?
    a) JLabel
    b) JTextField
    c) JButton

26. Swing components are _____.
    a) Lightweight
    b) Heavyweight
    c) Medium-weight

27. Which of the following is not a layout used with Java Swing?
    a) FlowLayout
    b) BorderLayout
    c) LinearLayout

28. In the Java coordinate system _____.
    a) x increases towards the right, y increases as you move down.
    b) x decreases towards the right, y decreases as you move down.
    c) x increases towards the right, y increases as you move up.

29. When using the Observer Design Pattern with Java's built in support your Subject subclasses _____.
    a) java.util.Observer
    b) java.util.Observable
    c) java.util.Subject

30. In the Model View Controller Design Pattern what component typically takes on the role of interacting with the user?
    a) Model
    b) View
    c) Controller

Part 2: Short Answer Questions 40 points [points vary per question, spend time wisely]

1. Given the code samples from classes Person and PersonBuilder write the constructor inside Person needed to complete the builder design pattern. [7 points]

```
public class PersonBuilder{
    private int personId;
    private String firstName;
    private String email;
    public int getPersonID(){ return personId; }
    public String getFirstName(){ return firstName; }
    public String getEmail(){ return email; }
    private PersonBuilder() { }
    public static PersonBuilder create() { return new PersonBuilder(); }
    public PersonBuilder personId (int personId)     { this.personId = personId;   return this; }
    public PersonBuilder firstName(String firstName) { this.firstName = firstName; return this; }
    public PersonBuilder email    (String email)     { this.email = email;         return this; }

    public Person build() { return new Person(this); } // Call to the constructor
}
public class Person{
    private int personId;
    private String firstName;
    private String email;

    public int getPersonID(){ return personId; }
    public void setPersonID(int personID){this.personId = personID; }

    public String getFirstName(){ return firstName; }
    public void setLirstName(String firstName){this.firstName = firstName;}

    public String getEmail(){ return email; }
    public void setEmail(String email){this.email = email;}

// TO DO: Write Constructor to complete Builder Design Pattern
```

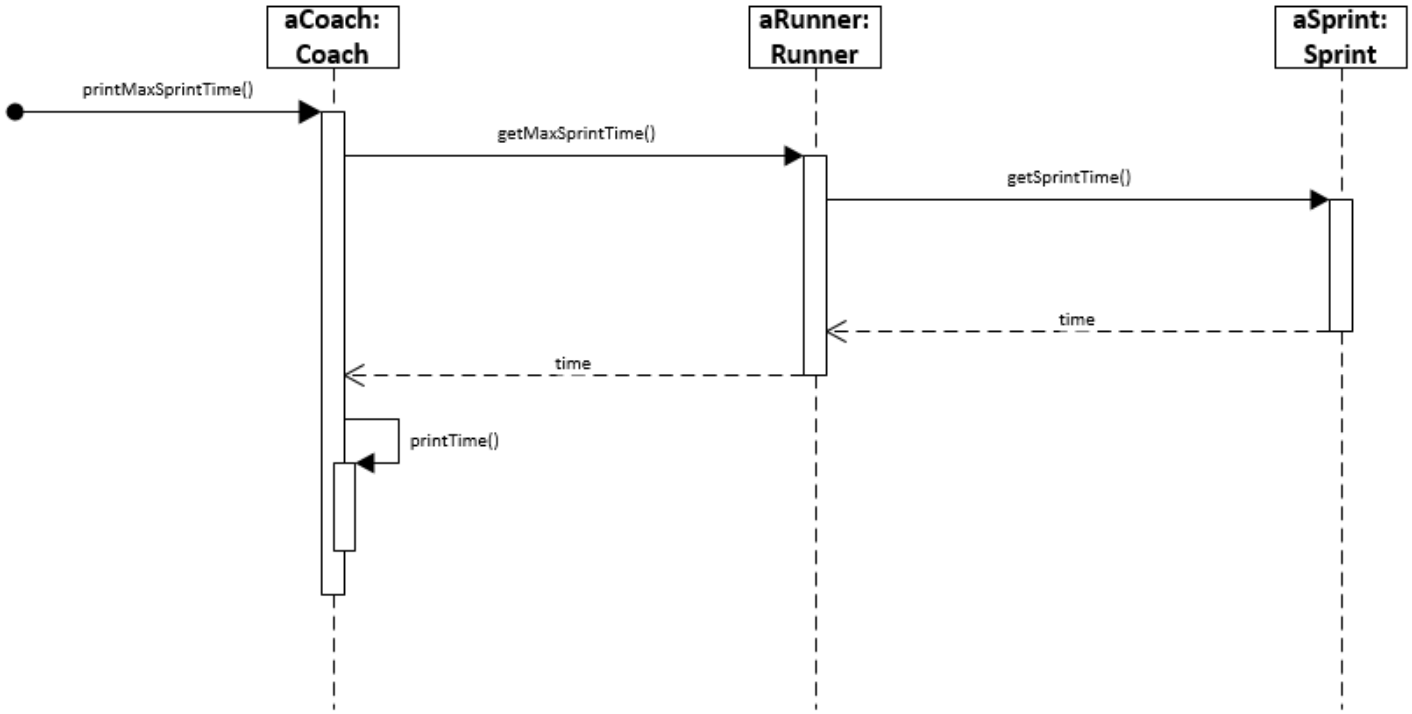
// Answer:

```
    public Person(PersonBuilder builder)
    {
        personId = builder.getPersonID();
        firstName = builder.getFirstName();
        email = builder.getEmail();
    }
```

2. Create a UML Sequence Diagram based on the following class names, you may select method names that make sense to complete the diagram. [7 points]

Classes: Coach, Runner, Sprint

Scenario: The three objects interacting in the scenario have all been instantiated. The coach needs to retrieve the maximum sprint time, in seconds, from runner and then print this value to the console.



3. Matching: Examine the code samples below, what is the Big-O for each? [6 points]

Possible Big-O: O(1) O(n) O(n^2) O(log n)

Each Big-O is only used once, only 3 are needed

i)
```
for(int i = 1; i < n; i++){}
```

Big-O is (answer here) _____**O(n)**

ii)
```
for(int i = 1; i < n; i = i * 2){}
```

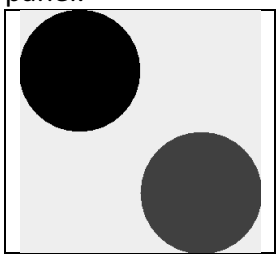
Big-O is (answer here) _____**O(log n)**

iii)
```
for(int i = 1; i < n; i++){
    for(int k = 1; k < n; k++){}
}
```

Big-O is (answer here) _____ **O(n^2)**

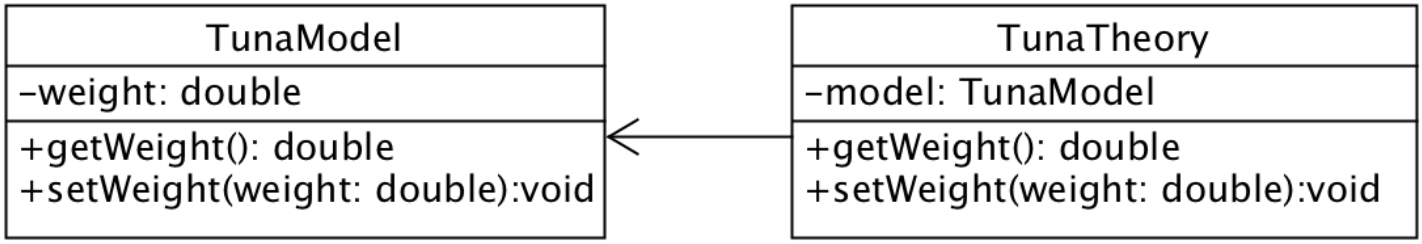
4. Given the screen shot and starter code, write the needed Java 2D code to complete the image. [10 points]

- The Panel is 300 pixels wide and 300 pixels tall.
- The upper circle in Black is drawn from the upper left corner of the panel and has a diameter ½ the width of the panel
- The lower circle in DarkGray is drawn from the middle of the panel and has a diameter ½ the width of the panel.



```
import javax.swing.JPanel;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Dimension;
public class A extends JPanel {
    public A(){
        super();
        setPreferredSize(new Dimension(300,300));
    }
    @Override
    public void paintComponent(Graphics g){
      // Calculate the needed values
      // (w is width, h is height, x is x-coordinate, y is y-coordinate)

        int wUpper = 300 / 2; // one half width of panel

        int hUpper = 300 / 2; // one half height of panel

        int xUpper = 0; // 0,0 is for upper top-left corner

        int yUpper = 0;


        int wLower = 300 / 2;

        int hLower = 300 / 2;

        int xLower = 300 / 2; // calculate x as one half of width

        int yLower = 300 / 2; // calculate y as one half of height

        g.setColor(Color.black);
        g.fillOval(xUpper, yUpper, wUpper, hUpper);
        g.setColor(Color.darkGray);
        g.fillOval(xLower, yLower, wLower, hLower);
    }
}
```

5. Use the UML below to write the code for a simple delegate design pattern (both classes) [10 points]




```
public class TunaModel{
    private double weight;
    public double getWeight(){ return weight; }
    public void setWeight(double weight){ this. weight = weight; }
}

public class TunaTheory{
    private TunaModel model = new TunaModel();
    public double getWeight(){ return model.getWeight(); }
    public void setWeight(double weight){ model.setWeight(weight); }
}
```

Scrap Paper, Name: _____

Scrap Paper, Name: _____

Full Name: _____

CST8288 15F Multiple Choice Question Answer Sheet: Term Test A

Note: Question numbers are vertical down the page

Question Number	Letter Answer
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Question Number	Letter Answer
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	

Question Number	Letter Answer
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	