

CMSC132
Fall 2005
Midterm #1

Grader Use Only:

#1		(21)
#2		(29)
#3		(21)
#4		(18)
#5		(11)
Total		(100)
Honors		(10)

First Name: _____

Last Name: _____

Student ID: _____

Section time _____ **TA:** _____

I pledge on my honor that I have not given or received any unauthorized assistance on this examination.

Your signature: _____

General Rules (Read):

- This exam is closed book and closed notes.
- If you have a question, please raise your hand.
- Total point value is 100 points.
- Answer True/False questions by circling the T or F at the end of the question.
- Answer multiple-choice questions by circling the letter (e.g., a, b) at the front of each choice.
- Answer essay questions concisely using 1 or 2 sentences. Longer answers are not necessary and are discouraged.
- **WRITE NEATLY.** If we cannot understand your answer, we will not grade it (i.e., 0 credit).

Problem 1 Software Development & Testing (21 pts)

- A. (1 pt) Why are large pieces of software harder to develop than small pieces of software?
- a. Larger software take much more time to write
 - b. Larger software take many more programmers to write
 - c. Larger software take much faster computers to execute
 - d. Larger software require more computer memory
 - e. Larger software are more complex
- B. (1 pt) The software life cycle is a sequence of essential operations necessary for producing quality software. Which of the following operation is not part of the software life cycle?
- a. Specification
 - b. Coding
 - c. Debugging
 - d. Testing
 - e. Release
- C. (4 pts) The waterfall model of software development...
- a. Describe its basic approach
 - b. Should be used for large software projects T or F
 - c. Is accepted as the best model for developing software T or F
- D. (4 pts) The unified model of software development...
- a. Describe its basic approach
 - b. Should be used for large software projects T or F
 - c. Is accepted as the best model for developing software T or F
- E. (5 pts) Problem specification is a phase of the software life cycle
- a. Describe its basic goal
 - b. Specifications usually never change T or F
 - c. Specifications can describe all possible inputs T or F
 - d. Specifications written in English are very precise T or F

F. (6 pts) Program testing

a. What is “black box” testing?

b. Empirical testing depends on analysis of code

T or F

c. Empirical testing can eliminate all errors

T or F

d. Program verification depends on calculating pre & post-conditions

T or F

e. Good unit tests eliminate the need for integration tests

T or F

Problem 2 Object-Oriented Design (29 pts)

G. (5 pts) Procedural-oriented view vs. object-oriented design

a. Describe the procedural-view approach to software design

b. Procedural-oriented view produces programs which run faster

T or F

c. Why do modern software developers prefer object-oriented design?

H. (6 pts) Abstraction and encapsulation

a. Why do software developers use abstraction and encapsulation?

b. Give an example of Java support for abstraction

c. Give an example of Java support for encapsulation

I. (6 pts) Inheritance

a. Inheritance describes a relationship between related classes

T or F

b. Inheritance encourages code reuse

T or F

c. Many forms of inheritance exist in Java

T or F

d. Java supports combined (multiple) inheritance

T or F

e. Provide an example of how *specification* is supported in Java

- J. (12 pts) Given the following problem description, produce an object-oriented solution. Include as many details as possible. Draw a UML class diagram (you may write code for Java classes ONLY if you don't know UML, but will lose points if you do so).

Write a submit server for student programming projects. The submit server supports two types of users, students and TAs. Students can submit code to the server. The server must record the time and code for student submissions in a database. Once submitted, the server must compile code execute it using test cases as input. Test cases may be public or private. Scores from executing code for each test case must be recorded and saved in the database. Students may get their scores for each test case from the server. TAs can also obtain test scores for all student submissions. In addition TAs can also examine code from all student submissions.

Problem 3 Unified Modeling Language (21 pts)

K. (6 pts) UML ...

- a. Provides a software blueprint for object-oriented software systems T or F
- b. Can describes both the static and dynamic behavior of a software system T or F
- c. Name two types of UML diagrams (excluding class diagrams)

- d. Name four types of relationships between classes in a UML class diagram

L. (8 pts) Given the following Java code, draw its UML class diagram (you may write code for Java classes ONLY if you don't know UML). Include as much information as possible in the UML class diagram.

```
public interface Device {
    public void setPower(double power);
}

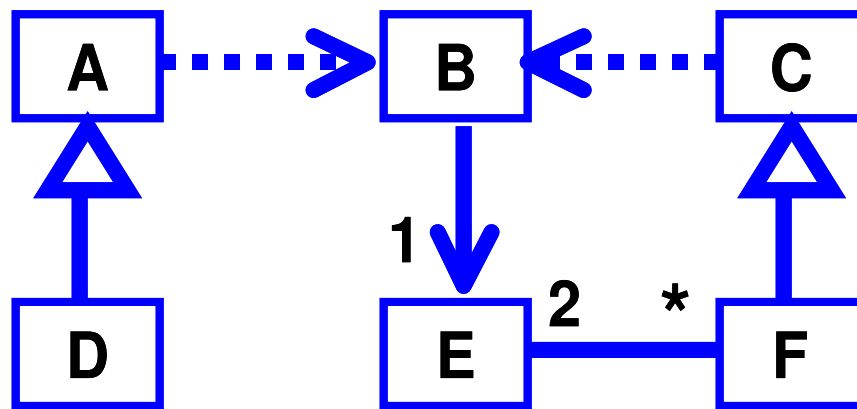
public class Speaker {
    int volume = 1;
    public void setVolume(int volume) {
        this.volume = volume;
    }
    public String toString() {
        return "Volume: " + volume;
    }
}

public class TV {
    protected Speaker[] speaker;
    public TV() {
        speaker = new Speaker[2];
        speaker[0] = new Speaker();
        speaker[1] = new Speaker();
    }
    public Speaker getSpeaker(int number) {
        return speaker[number-1];
    }
}

public class HDTV extends TV implements Device {
    private double power;
    public HDTV() {
        speaker[0].setVolume(10);
        speaker[1].setVolume(20);
        power = 120;
    }
    public void setPower(double power) {
        this.power = power;
    }
    public String toString() {
        String res = speaker[0] + " " + speaker[1];
        return (res + " Power: " + power);
    }
}
```


M. (7 pts) Consider the UML diagram below:

- a. Which class contains class F?
- b. Which class uses class B?
- c. Which class may change if class E changes?
- d. How many instances of class F does class E have?
- e. How many instances of class B does class E have?
- f. Class D may be used wherever Class A is used T or F
- g. Class C may be used wherever Class F is used T or F



Problem 4 Java Programming (18 pts)

N. (7 pts) Java Language Features

- a. Static initialization blocks are all executed before constructors T or F
- b. Non-static initialization blocks are all executed before constructors T or F
- c. Visibility modifiers determine which variables may be used where in Java T or F
- d. List the four types of visibility modifiers in Java

O. The class Laptop is defined as follows:

```
public class Laptop implements Comparable {
    static ArrayList allLaptops = new ArrayList();
    private int id;

    public Laptop(int id) {
        this.id = id;
        allLaptops.add(this);
    }
    public String toString() {
        return id + " ";
    }
}
```

- a. (4 pts) Write code for the class Laptop so that it implements the Comparable interface in a way that larger id numbers are considered to be larger.

```
    public int compareTo(Object o) {
```

```
}
```

- b. (4 pts) Write code for a new method for class Laptop that will shuffle Laptop objects present in the ArrayList object associated with the static reference variable **allLaptops** (hint, use a method available in the Java Collections library). The method should then print out the list of laptops using System.out.println(). Use the new Java 1.5 for loop construct for printing the laptops (or else you will not receive full credit).

```
    public void shufflePrintLaptops() {
```

```
}
```

- c. (3 pts) Define an enumerated type named **LaptopType** representing two types of laptops: LowPower and HighPower. Modify the definition of class Laptop to add an instance variable (using the enumerated type) indicating the type of the laptop, and add code to the Laptop constructor to initialize all laptops to type HighPower by default.

```
public class Laptop implements Comparable {  
  
    // add enumerated type & new instance variable here  
  
  
  
  
  
  
  
  
  
    public Laptop(int id) {  
  
        // add initialization of instance variable to HighPower here  
  
  
  
  
  
  
  
  
  
    }  
  
}
```

Problem 5 Graphic User Interfaces (11 pts)

- P. (4 pts) In a graphics user interface (GUI)
- | | |
|---|--------|
| a. The controller is responsible for responding to events | T or F |
| b. The view should be able to access all the data in the model | T or F |
| c. The model can only be displayed by a single view at a time | T or F |
| d. Separating the GUI into model, view, controller improves its performance | T or F |

Q. (7 pts) Java Swing Code

Consider the following Java Swing code implementing a ButtonPanel class

```
public class ButtonPanel extends JPanel {
    private JButton jbutton;

    public ButtonPanel() {
        add(jbutton=new JButton("Test"));
        jbutton.addActionListener(new Handler());
    }

    class Handler implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            System.out.println("Action");
        }
    }
}
```

- a. (2 pts) In the ButtonPanel class, what represents the view?
- b. (2 pts) In the ButtonPanel class, what represents the controller?
- c. (3 pts) Write a new ButtonPanel constructor that uses an anonymous inner class to implement the same ActionListener for the JButton.

```
public class ButtonPanel extends JPanel {
    private JButton jbutton;

    public ButtonPanel() {
        add(jbutton=new JButton("Test"));
        jbutton.addActionListener(                                // finish adding code here

    }
}
```

Honors Section Problem (10 pts)

R. (3 pts) Out of the four types of relationships between classes in an UML class diagram...

a. Which represents “is a”?

b. Which represents “has a”?

c. Which represents “uses a”?

S. (2 pts) Write a small Java code fragment (less than 10 lines) that illustrates an infinite number of flow paths.

T. (5 pts) Write the Java code associated with the following UML diagram.

