NAME: _____ **STUDENT ID#:**_____
(Write your name on THIS PAGE ONLY) (Write your NEIU ID# on this and EVERY PAGE)

PRACTICE FINAL 4

CS 207, Programming II
Saturday, 201X
11:00 a.m. to 1:00 p.m.
Room LWH XXXX

EXAM PROBLEMS

Instructions:

1. **Do not turn this page until told to do so.**

2. This exam is ***closed book*** and ***closed notes***.

3. ***Write your STUDENT ID Number on every page***. If you do not write your ID Number on a certain page, you ***will not receive credit for the question on that page!***

4. There are **6** questions on the exam, one per page. Once you start, verify you have all 6 questions.

5. You must give your answer to a question on the ***same page on which the question appears***. If you put your answer on a different page, ***you will not receive any credit for it!***

6. For problems that ask you to write a ***method***, you must write the method header ***exactly*** as shown, but you do not need to write main( ).

7. For problems that ask you to write a ***program***, you should write the main( ) method ***only***—you can assume the following import statement and keyboard declarations are present.

```
import java.util.*;

Scanner keyboard = new Scanner (System.in);
or      Scanner kbd = new Scanner (System.in);
or      Scanner input = new Scanner (System.in);
```

8. For tracing problems, assume that any appropriate import statements are provided.

9. You may use "SOP" as an abbreviation for "System.out.print" and "SOPln" for "System.out.println".

10. You do not need to do any error checking of input values, ***unless the problem specifically asks you to do so!***

11. If you are caught looking at other papers or communicating with other students in any way, you will receive an **F** for the course.

STUDENT ID# _____

Problem 1 (XXX points).
Write the following program: CourseExamProcessing2.

You are given a data file named "student_exam_scores.txt", consisting of a list of exam scores for each student in a class. The scores are stored as decimal values, with either 4 or 5 scores per student. You do not know the number of scores in the list.

Your program is to do the following:

1. Read the scores into your program;
2. Display the number of students with 4 test scores and the number of students with 5 test scores;
3. Calculate, as a decimal value, the average of the scores;
4. Display the average for each student;
5. Handle the File I/O exception.

Below is a sample data set for the data file and the output generated from that data.

Sample course_exam_scores.txt file

```
87 88 85 83
96 94 95 92 91
73 77 76 74
```

Output generated for the above data file

```
/*PROGRAM OUTPUT



*/
```

STUDENT ID# _____

Problem 2 (XXX points)

Given the below interface and abstract class, implement the non-abstract class Dragon as follows:

1. Dragon should inherit from the Monster abstract class.

2. Dragon should take in the Dragon's name and color as parameters to its constructor and set the name, and store color as a private data field in Dragon. This field should have a public getter method as well.

3. Dragon should implement any required **methods** from the VideoGameCharacter interface and Monster super-class.

4. When the Dragon class' attack() method is called, it should output the Dragon's name concatenated with the color and " Dragon breathes fire!"

Supporting classes and application class output

```
public abstract class Monster implements VideoGameCharacter
{
   private String name;
   public Monster(String name)
   {
      this.name = name;
   }
   public String getName() {
      return name;
   }
}
```

```
public interface VideoGameCharacter
{
   public void attack();
   public String getName();
}
```

```
public class MonsterApp
{
   public static void main(String [] args)
   {
      Dragon fred = new Dragon("Fred", "Red");
      System.out.println(fred.getColor());
      fred.attack();
   }
}

/*PROGRAM OUTPUT
 "Red"
 "Fred the Red Dragon breaths fire!"
*/
```

STUDENT ID# _____

Problem 3 (XXX points)

Please write the exact output of class PlaneApp in the box provided on the corresponding page of the Answer Packet .

```
public class Plane
{
   private String name;
   private int callNumber;

   public Plane(String n)
   {
      this(n, 000);
   }

   public Plane(String n, int identifier){
      name = n;
      callNumber = identifier;
   }
   public String getName(){
      return name;
   }
   public int getCallNumber(){
      return callNumber;
   }
   public void setCallNumber(int cn){
      callNumber = cn;
   }
   public String toString(){
      displayProperties();
      return "Plane: " + name;
   }
   public void displayProperties(){
      System.out.println("One Classy Plane");
   }
}
```

```
public class CargoPlane extends Plane
{
   private String type ; //names: commercial, private

   public CargoPlane(){
      super("Bravo", 239); //name, call number
      type="commercial";
   }
   public CargoPlane(String n, String t){
      super(n, 727);
      type = t;
   }
   public String getType(){return(type);}

   public void displayProperties(){
      System.out.println(super.getCallNumber());
      System.out.println("Very Fast");
      super.displayProperties();
   }

   public boolean equals(Object o){
      System.out.println("Fancy Plane");
      CargoPlane cpThis = (CargoPlane)this;
      CargoPlane cpThat = (CargoPlane)o;

      if(!(o.getClass().equals(this.getClass())))
         System.out.println("crashing");
      else
         System.out.println("phew, not crashing");

      return(cpThis.getName().equals(cpThat.getName()));
   }
}
```

```
public class PlaneApp
{
   public static void main(String [] args){
      Plane p = null;
      Plane cc = new Plane("Coolness Cruiser");
      System.out.print("Plane p is: " );
      System.out.println(p);
      System.out.print("Plane cc is: ");
      System.out.println(cc);
      System.out.println("/************************************/");
      CargoPlane cp1 = new CargoPlane();
      CargoPlane cp2 = new CargoPlane("Dottie", "Commercial");
      Plane p1 = new CargoPlane("Chug", cp1.getType());
      Plane p2 = new Plane("Echo", 787);
      Plane [] planes = {p, cp1, cp2, p1, p2, null};
      System.out.println("/************************************/");
      planes[0] = cp2;
      System.out.println(cp1.equals(planes[0]));
      cp1.displayProperties();
      System.out.println(planes[0]);
      System.out.println("Fly the friendly skies");
      System.out.println("/************************************/");
   }
}
```

STUDENT ID# _____

Problem 4 (XXX points)

Write the following JAVA program: EnterCharacters.

The program is to prompt the user to enter 10 characters from the keyboard. The user may enter any character available from the keyboard, including alpha, numeric and special characters.

Once entered, the program will display the characters entered by the user as well as the number of times each character has been entered.

Considerations: only single characters may be entered (for example, "10" is invalid). Should a user attempt to enter a multi-character value, the program is to throw an InputMismatchException (which requires the java.util import)

You are given starter code. You may only use the variables listed in the variable declaration section.

To receive credit for this problem you may not add any additional variables to your solution.

```
import java.util.*;

public class EnterCharacters
{
   public static void main(String [] args)
   {
      //variable declarations
      Scanner kbd = new Scanner(System.in);
      int [] chars = new int[128];
      String s;
      char c;
      int counter = 1;
      int i;









 }
}
```

STUDENT ID# _____

Problem 5 (XXX points)

Please write the exact output of the program Tracing in the box provided on the corresponding page of the Answer Packet.

```
public class Tracing{
   public static void main(String[ ] args){
      int i = 7, j = 0, result = 0;
      int [] arr = { 92, 84, 17};

      while(j < 3){
         try
         {
            result = i/j;
            for(int row = 1; row < 4; row++)
               System.out.println("arr val: " + arr[row] + "\t");
         }
         catch(ArithmeticException e)
         {
            System.out.println("arithmetic error.");
         }
         catch(ArrayIndexOutOfBoundsException e)
         {
            System.out.println("Array index out of bounds");
         }
         catch(Exception e)
         {
            System.out.println("generic exception.");
         }
         finally
         {
            System.out.println("\nHowdy");
         }
         j++;
      }
      j = 0;
      try
      {
         result = i/j;
         arr[3] = 19;
      }
      catch(ArithmeticException e)
      {
         System.out.println("arithmetic error.");
      }
      catch(ArrayIndexOutOfBoundsException e)
      {
         System.out.println("Array index out of bounds");
      }
      catch(Exception e)
      {
         System.out.println("generic exception.");
      }
      finally
      {
         System.out.println("Done");
      }
   }
}
```

Problem 6 (XX points) **STUDENT ID#** _____

Please write the exact output of class BallApp found on Answers Page in the corresponding box on that page.

```
1  public class Ball
2  {
3     private int velocity;
4
5     public Ball()
6     {
7        System.out.println("BALL, line 7: No-arg Ball constructor");
8        velocity = 1;
9     }
10
11    public Ball(int x)
12    {
13       this(x, 5);
14       System.out.println("BALL, line 14: One-arg Ball constructor");
15       System.out.println("BALL, line 15: One-arg Ball constructor velocity = " + this.velocity);
16    }
17    public Ball(int x, int y)
18    {
19       System.out.println("BALL, line 19: Two-Arg Ball constructor");
20       velocity = x;
21       System.out.println("BALL, line 21: Two-Arg Ball constructor y is now: " + y);
22    }
23
24    public void setVelocity(int velocity)
25    {
26       this.velocity = velocity;
27       System.out.println("BALL, line 27: setVelocity. Velocity is: " + this.velocity);
28    }
29
30    public void hit()
31    {
32       System.out.println("BALL, line 32: Ball says you hit the ball a mile!!!");
33    }
34    public int getVelocity()
35    {
36       System.out.println("BALL, line 36: getVelocity");
37       return(this.velocity);
38    }
39 }
```

```
1  public class Softball extends Ball
2  {
3     public Softball()
4     {
5        System.out.println("SOFTBALL, line 5: You called the Softball constructor.");
6     }
7     public void riseBall()
8     {
9        System.out.println("SOFTBALL, line 9: You called the riseBall method of the Softball class.");
10    }
11    public int getVelocity()
12    {
13       System.out.println("SOFTBALL, line 13: getVelocity");
14       return(super.getVelocity());
15    }
16 }
```

```
1  public class Baseball extends Ball
2  {
3     public Baseball()
4     {
5        System.out.println("BASEBALL, line 5: You called the Baseball constructor.");
6     }
7     public void hit()
8     {
9        System.out.println("BASEBALL, line 9: You tore the cover off!!!");
10    }
11    public void setVelocity(int velocity)
12    {
13       super.setVelocity(velocity);
14       System.out.println("BASEBALL, line 17: setVelocity. Velocity is: " + getVelocity());
15    }
16    public int getVelocity()
17    {
18       System.out.println("BASEBALL, line 18: getVelocity");
19       return(super.getVelocity());
20    }
21 }
```

Problem 7 (XX points) **STUDENT ID#** _____

The following eliminate(String, char, char, char) is to return a new string that is composed of the original string less the three character arguments.

As written, code works as expected with one unacceptable point.

To solve this problem you are to do the following:

1. Identify the issue;
2. Rewrite the given code (you are not to write an entirely different solution) so that the issue is corrected.

Note: the output for the given sample string should be the following: marmarquite . For your corrected solution ONLY the given output string is to be returned. No credit will be given if the returned value is not exact or contains additional characters, for example a leading space.

```
1  public class Eliminate
2  {
3     public static void main(String [] args)
4     {
5        System.out.println(eliminate("mary mary quite", 'y', ' ', 'q'));
6     }
7     public static String eliminate(String str, char c1, char c2, char c3)
8     {
9        String newString = null;  //store original string less removed characters
10       int i = 0;              //loop counter
11
12       for(i = 0; i < str.length(); i++)
13       {
14          if(str.charAt(i) == c1 || str.charAt(i) == c2 || str.charAt(i) == c1)
15          {
16             //do nothing
17          }
18          else
19          {
20             newString = newString + str.charAt(i); //add to newString
21             //newString = str.charAt(i) + newString; //This ordering results in "etiuqramram"
22          }
23       }
24       return(newString);
25    }
26 }
```