Midterm Examination Problem Sheet
TIME: 04/18/2009, 19:00–21:00

This is a open-textbook exam. You can use the "Absolute Java" textbook as your reference during the exam. Any other references are not allowed. Any form of cheating, lying or plagiarism will not be tolerated. Students can get zero scores and/or fail the class and/or be kicked out of school and/or receive other punishments for those kinds of misconducts.
Both English and Traditional Chinese (if suited) are allowed for answering the questions. We do not accept any other languages.

1 Java Night Countdown (30%)

Before the coming of Java night, people are so excited that they want to know the number of minutes until the holy event. Thus, we should design a countdown program for the purpose. Please answer the following questions after reading the code. You can assume that the event date/time is always later than now.

```
1  //Countdown.java
2  package javanight.util;
3  import java.util.*;
4
5  public class Countdown{
6    int nDay, nHour, nMinute;
7
8    Countdown(int year, int month, int day, int hour, int min){
9      Date today = new Date();
10     Date goal = (new GregorianCalendar(year,month-1,day,hour,min)).getTime();
11     int diff = (int)((goal.getTime() - today.getTime()) / (1000 * 60));
12     nDay = diff / (60 * 24);
13     diff %= (60 * 24);
14     nHour = diff / (60);
15     diff %= (60);
16     nMinute = diff;
17   }
18   boolean countdown(){ return decrease_minute(); }
19   boolean decrease_day(){
20     if (nDay > 0) { nDay--; return true; }
21     else return false;
22   }
23   boolean decrease_hour(){
24     if (nHour > 0){ nHour--; return true; }
25     else if (decrease_day()){ nHour = 23; return true; }
26     else return false;
27   }
28   boolean decrease_minute(){
29     //(2)
30   }
31   int get_total_minutes(){
32     //(3)
33   }
34   int get_minutes(){ return nMinute; }
35   int get_hours(){ return nHour; }
36   int get_days(){ return nDay; }
37 }
```

```
1  //Timer.java
2  package javanight.publicity;
3  //(4)
4
5  public class Timer{
6    public static void main(String[] arg){
7      Countdown timer = new Countdown(2010, 4, 18, 19, 0);
8      do{
9        System.out.print(timer.get_total_minutes() + ",␣");
10       System.out.print(timer.get_minutes() + ",␣");
11       System.out.print(timer.get_hours() + ",␣");
12       System.out.println(timer.get_days());
13     }while(timer.countdown());
14   }
15 }
```

(1) (2%) What is the fully-qualified name of the class `Countdown`?

(2) (4%) Complete the method `decrease_minute` (marked with `//(2)` above). Yes, you need to "guess" the intended functionality.

(3) (4%) Complete the method `get_total_minutes` (marked with `//(3)` above). It returns the total number of minutes until the event. For example, if there are 2 days, 3 hours and 4 minutes left, the method returns 3064.

(4) (2%) When compiling the two files above, Java compiler laughs (hahaha) with

```
Timer.java:7: cannot find symbol
```

Adding a line around the place marked with `//(4)` can solve the problem. What is the line?

(5) (6%) The hahaha goes away after adding the line in the previous question, but new ones come! What are the minimum changes need to be made in `Countdown.java` to make all the hahaha go away? (*Hint: check access permissions*)

(6) (12%) Speaking of permissions, what is the tightest permission of each instance variables/methods/constructors of the class `Countdown` to make all the code above work? Note that there are 12 of them.

2 Java Night Choir (20%)

In Java night, all the 200 OOP students will sing as a choir (with You-Know-Who as the conductor). If there are six students in the order $(1, 2, 4, 3, 5, 6)$, we can try to arrange them in four rows, like

```
1
2
4 3
5 6
```

More specifically, if there are N students in R rows, the last $N\%R$ rows would contain $(N/R) + 1$ students[1], and the other rows would contain (N/R) students. The students would seat from the most top-left position, in row major, to the most bottom-right position. Finish the following code that arrange/rearrange the students to the choir positions (4% for each method)
The quality of your code may be taken into account when grading. You can assume that $N > R > 0$.

```
1  //Choir.java
2  public class Choir{
3    private int[][] position;
4    private int nStudent;
```

[1]Here '/' means integer division.

```
5     /** arrange the IDs to nRow rows */
6     public Choir(int[] IDs, int nRow){
7       //(1)
8     }
9     /** arrange the IDs to 3 rows */
10    public Choir(int[] IDs){
11      //(2)
12    }
13    /** re-arrange the internal positions */
14    public void reshape(int nRow){
15      //(3)
16    }
17    /** return the ID at position row-num,
18        for instance, position 2-1 of the case above returns 3
19        return -1 if no such position */
20    public int getID(int row, int num){
21      //(4)
22    }
23    /** show the choir
24        such that (1, 2, 4, 3, 5, 6) with 4 rows
25        would output as the case above */
26    public void showChoir(){
27      //(5)
28    }
29  }
```

3 Java Night Singer PK (20%)

The Java Night is so hot that all the singers are competing for a single five-minute slot of showing off! The organizing team has thus decided the following selection mechanism for picking the one: the first half of the candidates would first form a subgroup, compete among themselves and declare their representative; the second half would do the same. Then, the two representatives would have a one-on-one competition to declare the final winner. Given that this seems to be a fair way, the subgroups, sub-subgroups, ... all decide to do the same. Such a mechanism leads to the following Java code:

```
1   //PK.java
2   class Singer{
3     int ID;
4     int score;
5   }
6   public class PK{
7       static int count;
8       /* (1) */ swap(Singer a, Singer b){
9         Singer tmp = a;
10        a = b;
11        b = a;
12      }
13      /** after running, arr[left] should contain
14          the best singer between arr[left] and arr[right-1]
15          and will be returned */
16      private static Singer compete(Singer[] arr, int left, int right){
17          int len = right - left;
18          if (len > 1){
19              int middle = left+len/2;
20              Singer first = compete(arr, left, middle);
21              Singer second = compete(arr, middle, right);
22              if (first.score < second.score)
```

```
23                  swap(arr[left], arr[middle]);
24              count++;
25
26              System.out.print(count + ":");
27              for(int i = 0; i < arr.length; i++)
28                  System.out.printf(" %d(%d)", arr[i].ID, arr[i].score);
29              System.out.println();
30          }
31          return arr[left];
32      }
33      public static Singer compete(Singer[] arr){
34          count = 0;
35          return compete(arr, 0, arr.length);
36      }
37  }
```

(1) (4%) Fill in the part marked with `/* (1) */` with the tightest access permission modifier.

(2) (4%) The method `swap` does not work as expected. How would you change the code from lines 9 to 11 to make the method work?

(3) (4%) After correcting `swap`, what is the output when the public `compete` is called with an array containing $\{(1, 80), (2, 90)\}$, where $(1, 90)$ means a singer of ID 1 and score 90?

(4) (4%) After correcting `swap`, what is the output when the public `compete` is called with an array containing $\{(3, 90), (5, 80), (6, 70), (1, 95)\}$?

(5) (4%) After correcting `swap`, what is the output when the public `compete` is called with an array containing $\{(3, 90), (5, 80), (1, 85), (2, 60), (4, 70)\}$?

4 Java Night Drinks (30%)

What is the best drink for Java night? Java coffee, of course! Now that you are in charge of handling drinks in Java night, you'd better organize all the potential drinks you have. Let's define the following classes first:

```
1  public class Drink{
2    private double amount;
3    public void drink(double sip){
4      if (amount > sip) amount -= sip;
5      else amount = 0;
6    }
7    public double getAmount(){ return amount; }
8  }
9  class Java extends Drink{
10   double caffeine;
11   public Java(double a, double c){
12     amount = a;
13     caffeine = c;
14   }
15 }
16 class Milk extends Drink{ }
17 class FatlessMilk extends Milk{ }
18 class BlendedJava extends Java{
19   Milk milk;
20   double ratio;
21 }
```

(1) (2%) At least how many instance variables does class `BlendedJava` have?

(2) (3%) There is one place that may result in compile error (hahaha) in the constructor of class `Java`. What is it and how could you fix it by changing only one line?

(3) (3%) There is one place that may result in compile error (hahaha) in the constructor of class `BlendedJava`. What is it and how could you fix it?

(4) (4%) We want to modify the action of drinking in class `BlendedJava` as follows. For some `sip` values, you would drink `(ratio * sip)` from the Java coffee part and `(sip - ratio * sip)` from the Milk part. Write down such a `drink` method for class `BlendedJava`.

```
1  public class JavaDemo{
2    public static void main(String[] argv){
3      Milk m1 = new Milk();
4      Milk m2 = new Milk();
5      Java j1 = new Java(3.0, 2.0);
6      BlendedJava b1 = new BlendedJava();
7      BlendedJava b2 = new BlendedJava();
8      b1.milk = m1;
9      j1 = b1;
10     b1.milk = new FatlessMilk();
11     Milk[] marr = new Milk[3];
12     marr[0] = b1.milk;
13     marr[1] = m2;
14     /* CODE */
15   }
16 }
```

Now, consider another java source file above. Please write down the output when the `/* CODE */` part is replaced by the following lines (respectively for each subproblem). If you think there is a compile error, write "compile error" or "hahaha." If you think there is a run-time error (exception), write down "run-time error" or "ohohoh."

(5) (2%) `System.out.println(m1 instanceof FatlessMilk);`

(6) (2%) `System.out.println(marr instanceof Milk);`

(7) (2%) `System.out.println(marr[1] == marr[0]);`

(8) (2%) `System.out.println(marr[2].getAmount());`

(9) (2%) `System.out.println(b1 instanceof Object);`

(10) (2%) `System.out.println(j1 instanceof BlendedJava);`

(11) (2%) `System.out.println(b2.milk);`

(12) (2%) `System.out.println(marr[0] instanceof BlendedJava);`

(13) (2%) `System.out.println(j1.caffeine == b1.caffeine);`

5 Brainstorming Time

(1) (Bonus 5%) On You-Know-Who's cellphone, a program shows the following message:

```
Null Pointer
java/lang/NullPointerException
```

What might be the cause of the message?