NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING
EXAMINATION FOR
Semester 1: 2005/6

CS1101X – PROGRAMMING METHODOLOGY

November 2005  Time allowed: 2 hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper consists of SIX (6) questions and comprises THIRTEEN (13) printed pages.
2. This is an OPEN BOOK examination.
3. Answer all questions. You may use pencil to write your codes. Pen is preferred for other questions.
4. Fill in your Matriculation Number clearly below.
5. You must submit only this question paper and no other document.

Matriculation number: 

<table>
<thead>
<tr>
<th>Question</th>
<th>Total</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100</strong></td>
<td></td>
</tr>
</tbody>
</table>
1. This year, 2005, is Java’s __________ anniversary. [1 mark]

2. [15 marks]
   Study the following program carefully, and write out the output. There is no error in the program.

   ```java
   class Confusing {
       public int X (int n) {
           return n/2;
       }
       public int Y (int n, int m) {
           return m - n;
       }
   }
   class TestConfusing {
       public static void main( String[] args ) {
           Confusing n = new Confusing();
           Confusing m = new Confusing();
           int x = n.X(25);
           int X = n.X(3) + m.X(3);
           int y = n.Y(x, X-2);
           int Y = n.Y(x, m.Y(10,20));

           int sum = 0;
           for ( int i = 1; i <= 5; i++ ) {
               sum = (int) Math.pow(2,i) - n.Y(sum, i);
           }

           System.out.println ("x = " + x);
           System.out.println ("X = " + X);
           System.out.println ("y = " + y);
           System.out.println ("Y = " + Y);
           System.out.println ("sum = " + sum);
           if (sum >= Y)
               if (x > y)
                   x = n.X(y);
               else if (X > Y)
                   X = n.X(sum - Y);
               else
                   Y = n.Y(x,y);
           sum = n.Y(n.X(x), n.Y(y, y));

           System.out.println ("x = " + x);
           System.out.println ("X = " + X);
           System.out.println ("y = " + y);
           System.out.println ("Y = " + Y);
           System.out.println ("sum = " + sum);
       }
   }
   ```

Write your answers here.

```java
   x = 
   X = 
   y = 
   Y = 
   sum = 
   x = 
   X = 
   y = 
   Y = 
   sum = 
```
3. [18 marks]
Study the following program carefully and identify all the invalid statements (syntax error, logic error, etc.) by their line numbers. For each invalid statement, you should state the reason for the error. Note that there are no more than 10 errors.

1. public class Electronics {
2.     private static final double UNKNOWN = -1.0;
3.     public static int quantity;
4.     public String brand;
5.     protected String color;
6.     private float price;
7.     public Electronics (String brand) {
8.         this.brand = new String (brand);
9.         this.price = UNKNOWN;
10.    }
11.    protected String getColor (MP3Player mp3) {
12.         return color;
13.    }
14.    public double getPrice () {
15.         return price;
16.    }
17.    public void setPrice (float price) {
18.         this.price = price;
19.    }
20.    public static void incQty (int n) {
21.        quantity += n;
22.    }
23. }
24. public class MP3Player extends Electronics {
25.     final String DEFAULT_FORMAT = "WAV";
26.     private String audioformat;
27.     public MP3Player(String brand; String format) {
28.         super (brand);
29.         audioformat = format;
30.         incQty(1);
31.     }
32.     public MP3Player( MP3Player mp3 ) {
33.         this (mp3.brand, mp3.format);
34.         this.color = new String( mp3.color );
35.         this.price = mp3.price;
36.     }
37.     public boolean comparePrice (MP3Player mp3) {

double cost1 = mp3.getPrice();
double cost2 = getPrice();
if (cost2 > cost1) return true;
else return false;
}
}

class TestMain {
    public static void main( String[] args ) {
        Electronics myItem;
        MP3Player myMP3, herMP3;
        double myCost, herCost;
        int quantity = 0;

        myItem = new Electronics("Sony");
        myItem.color = "Silver";
        myItem.price = 250.99;
        myItem.quantity++;

        myMP3 = new MP3Player ((MP3Player) myItem);
        myMP3.incQty(1);

        herMP3 = new MP3Player("Apple", "WAV");
        quantity++;

        myCost = myMP3.getPrice();
        herCost = herMP3.getPrice();

        if (myCost > herCost &&
                myMP3.brand == herMP3.brand) {
            increaseCost (herMP3);
        }

        System.out.println (myItem.getColor());
        System.out.println (myItem.getPrice());
        System.out.println (myMP3.comparePrice(herMP3));
    }

    private void increaseCost (MP3Player mp3) {
        double price;
        price = mp3.getPrice();
        mp3.setPrice(price*2);
    }
}
Write your answers for Question 3 here.
The following method compares two input points p and q, both objects of the Point class. Note that x and y are the coordinates of a Point object, and they are integer values. Moreover, we assume here that the coordinates are non-negative.

```java
public static boolean lessThan (Point p, Point q) {
    if (p.x == q.x && p.y == q.y) return false;
    int x, y;
    for (int i=0; true; i++) { // outer loop
        x = 0;
        y = i;
        for (int j=0; j <= i; j++) { // inner loop
            if (x == p.x && y == p.y) return true;
            if (x == q.x && y == q.y) return false;
            x++;
            y--;
        }
    }
}
```

The number of iterations (outer and inner loops) is proportional to the values of x and y in p and q. This is not effective (as you can recall the example of GCD given in the lecture). Answer the following parts.

a. If p = (3, 2) and q = (4, 5), what is the output of `lessThan(p, q)`? Answer: _______

b. If p = (6, 3) and q = (4, 5), what is the output of `lessThan(p, q)`? Answer: _______

c. Rewrite the above `lessThan` method to remove all loops. Write your answer below.
5.  [16 marks]
We have a 2D array of integers:

```java
int[][] number = new int[6][6];
```

Its content is given in the table below:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>22</td>
<td>8</td>
<td>11</td>
<td>59</td>
<td>44</td>
</tr>
<tr>
<td>50</td>
<td>64</td>
<td>29</td>
<td>10</td>
<td>34</td>
<td>20</td>
</tr>
<tr>
<td>17</td>
<td>86</td>
<td>27</td>
<td>98</td>
<td>57</td>
<td>21</td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td>88</td>
<td>42</td>
<td>36</td>
<td>45</td>
</tr>
<tr>
<td>31</td>
<td>26</td>
<td>12</td>
<td>23</td>
<td>82</td>
<td>54</td>
</tr>
<tr>
<td>28</td>
<td>66</td>
<td>58</td>
<td>69</td>
<td>41</td>
<td>1</td>
</tr>
</tbody>
</table>

Consider the following code fragment, where selectionSort is the Selection Sort technique covered in lecture.

```java
for (int i = 0; i < 6; i++)
    selectionSort( number[i] );

// At this point, fill in table (a) on the content in the number array

int[] tempNumber = new int[6];
for (int j = 0; j < 6; j++) {
    for (int i = 0; i < 6; i++)
        tempNumber[i] = number[i][j];
    selectionSort( tempNumber );
    for (int i = 0; i < 6; i++)
        number[i][j] = tempNumber[i];
}

// At this point, fill in table (b) on the content in the number array
```

**Table (a) [5 marks]**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table (b) [5 marks]**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Due to the special property of the number array – a hint to answering to part (b) in the above, we can design an algorithm similar to binary search algorithm to search for a key in the number array. Below is the recursive algorithm with some parts (6 blank lines) left for you to fill. The method is to be invoked by calling:

```
search (wantedNumber, number, 0, 0, 5, 5)
```

which returns true if wantedNumber is in the number array, or false otherwise.

[6 marks]

```java
static boolean search (int key, int[][] table, int startX, int startY, int endX, int endY) {
    if (startX > endX || startY > endY) return ______ ;
    int midX = (startX + endX)/2;
    int midY = (startY + endY)/2;
    if (key == table[midX][midY]) return ______ ;
    if (key < table[midX][midY])
        return  __________________________________________
                __________________________________________ ;
    else
        return  __________________________________________
                __________________________________________ ;
}
```
6. **Pig Latin** [33 marks]

*Pig Latin* is a language game primarily used in English. It is usually used by children, who will often use it to converse in (perceived) privacy from adults, or simply for amusement. (From *Wikipedia.*)

We shall study one simplified version of Pig Latin here. It involves two cases. You may assume that words consist of only lowercase letters.

**Case 1: Words beginning with a consonant.**

If the word has one or more consonants before the first vowel in it, these consonants are moved to the end of the word, with their order maintained (example: “scratch” becomes “atchscr”). Then we add “ay” to the end of the newly formed word (example: “scratch” becomes “atchscr”, then it becomes “atchscray”).

More examples:

<table>
<thead>
<tr>
<th>English word</th>
<th>Pig Latin</th>
</tr>
</thead>
<tbody>
<tr>
<td>ball</td>
<td>allbay</td>
</tr>
<tr>
<td>yes</td>
<td>esyay</td>
</tr>
<tr>
<td>busy</td>
<td>usybay</td>
</tr>
<tr>
<td>pure</td>
<td>urepay</td>
</tr>
<tr>
<td>green</td>
<td>eengray</td>
</tr>
<tr>
<td>scratch</td>
<td>atchscray</td>
</tr>
</tbody>
</table>

**Case 2: Words beginning with a vowel.**

If the word begins with a vowel (a, e, i, o, u), add “y” before the word. Moreover, if the word ends with a consonant, add “ay” to the end of the word; if it ends with a vowel, add “yay” to the end of the word instead. Examples:

<table>
<thead>
<tr>
<th>English word</th>
<th>Pig Latin</th>
</tr>
</thead>
<tbody>
<tr>
<td>in</td>
<td>yinay</td>
</tr>
<tr>
<td>after</td>
<td>yafteray</td>
</tr>
<tr>
<td>echo</td>
<td>yechoyay</td>
</tr>
<tr>
<td>orange</td>
<td>yorangeyay</td>
</tr>
</tbody>
</table>

(a) Write a Java program **PigLatin.java** that defines the **PigLatin** class. The program should include the `convertPigLatin()` method to convert a word into its Pig Latin equivalent. The partial code is given below. You may add other method(s). [12 marks]

```java
public class PigLatin {
    private String word;

    public PigLatin(String word) {
        this.word = new String(word);
    }

    public String convertPigLatin() {
        // code to be completed
    }
}
```
Write your answer for Question 6 part (a) here. You need only fill in the `convertPigLatin()` method and other methods of your own here.
(b) Write a Java program `DemoPigLatin.java` that reads lines of text, converts every word into its Pig Latin equivalent and displays the converted text. A sample run is shown below.

```
$ java DemoPigLatin
Enter text: The quick brown fox...
Converted : eThay uickqay ownbray oxfay...

Enter text: jumps over the lazy dog.
Converted : umpsjay yoveray ethay azylay ogday.

Enter text: Oei!Get into the box!
Converted : yOeiyay!etGay yintoay ethay oxbay!

Enter text: This can’t be er... true
Converted : isThay ancay’tay ebay yeray... uetray

Enter text: or, is it? ABC?
Converted : yoray, yisay yitay? yABCay?

Enter text: <end-of-input>
```

Write your answer for Question 6 part (b) here.
Continue your answer for Question 6 part (b) here if you need to.
(c) Suppose the rule for words that begin with consonant remains, but the rule for words that begin with vowel is simplified as follows: for a word that begins with a vowel, simply add “ay” to the end of the word.

<table>
<thead>
<tr>
<th>English word</th>
<th>Pig Latin</th>
</tr>
</thead>
<tbody>
<tr>
<td>in</td>
<td>inay</td>
</tr>
<tr>
<td>after</td>
<td>afteray</td>
</tr>
<tr>
<td>echo</td>
<td>echoay</td>
</tr>
<tr>
<td>orange</td>
<td>orangeay</td>
</tr>
</tbody>
</table>

Replace the method `convertPigLatin()` in part (a) with a recursive version. You may change the signature of the method if you wish. [8 marks]

Write your answer for Question 6 part (c) here.