The University College of the Cariboo
Computing Science Department

COURSE: COMP 113 **SEMESTER:** Fall 2004

TERM TEST #1 *Solutions*

Total Marks: 20 marks

[1 mark]

1. What does the Java compiler do?

   the java compiler translates a source code file (.java) into a byte code file (.class), which can then be interpreted by the java virtual machine into the machine language for the CPU and operating system.

[1 mark]

2. The purpose of the following segment is to display the user's input string in reverse. The compiler does not produce any errors, but the program crashes each time it runs, indicating a problem at the output step in the **for**-loop. *Fix it!*

```
String in = "";
in = Input ("Enter a string:");

for (int i = in.length(); i>=0; i--)
   Output ( in.charAt(i) );
```

   the problem is an "off-by-one" error based on the index of the last character in a string, see the correction:

```
for (int i = in.length()-1; i>=0; i--)
```

[2 marks]

3. Write the program segment that inputs 3 numbers and displays them in sorted order.
   It does not matter how the numbers are input (one at a time, or together in a single string).
   For example, if the input is: 4, 9, 3, the output is: 3, 4, 9.

```
int a=0, b=0, c=0;          // input values
int low=0, mid=0, high=0;   // ordered values

a = Input ("Enter the first value:");
b = Input ("Enter the second value:");
c = Input ("Enter the third value:");

  // the following may seem long, but it is understandable...or should be!
if ((a <= b) && (a <= c))  // if a is the smallest
  if (b <= c)              // if b is second smallest
  {
    low=a;  mid=b;  high=c;
  }
  else                     // else, c is the second smallest
  {
    low=a;  mid=c;  high=b;
  }
```

```
else if ((b <= a) && (b <= c)) // a not smallest; check if b is the smallest
  if (a <= c)                  // if a is the second smallest
  {
     low=b;  mid=a;  high=c;
  }
  else                         // else, c is the second smallest
  {
     low=b;  mid=c;  high=c;
  }
else  // b not smallest; c *must* the smallest
  if (a <= b)                  // if a is the second smallest
  {
     low=c;  mid=a;  high=b;
  }
  else                         // else, b is the second smallest
  {
     low=c;  mid=b;  high=a;
  }

Output ("The order of values: "+low+", "+mid+", "+high);
```

[2 marks]

4. Write the program segment that translates a letter grade, stored in the char variable **letter**, into a number grade, assigned to the **int** variable **grade**. The possible letter grades are A, B, C, D, F.
   The corresponding number grade will be 4, 3, 2, 1, 0. You must use a **switch** statement.

```
switch (letter)
{
  case 'A':
    grade = 4;
    break;

  case 'B':
    grade = 3;
    break;

  case 'C':
    grade = 2;
    break;

  case 'D':
    grade = 1;
    break;

  case 'F':                  // note: a default option is not required
    grade = 0;
    break;
}
```

[3 marks]

5. What are the exact values calculated by Java in the calculations below? Indicate any that result in an error. In the second column, write the primitive data type of the resulting value for each assignment.

```
int     m = 0,   p = 12,   q = 7;
double  r = 40.0,  s = 3.0;
boolean x = false, y = false, z;
```

		value	datatype
a) = p / q;		1	int
b) = q / s		2.333	double
c) m = r + 1.1;	m =	*error*: int = double	--
d) q = 24.0 / p;	q =	*error*: int = double	--
e) = p % q;		5	int
f) z = (x || y) && !y;	z =	false	boolean

[2 marks]

6. The following loop segments compile and run, but both have errors. Identify the problems and fix them!

corrections are **bold and underline**

```
a) int sum=0;

     // loop & sum values from 1 to 10
   for (int i=1 ; i<=10; i++)
   {
      sum = sum + i;
      i++;
   }
   Output ("The sum is: " + sum);
```

```
b) int input=0, total=0;

    // loop until user inputs -99
   while (input != -99)
   {
      total = total + input;
      input = Input ("Next value :";
      total = total + input;
   }

   Output ("Total is "+total);
```

[2 marks]

7. Use either **for, while,** or **do_while**.
   Write the code segment that asks the user for the number of values that will be entered (**n**). It then repeats for **n**-times, asking the user for a value each time.
   At the end, the maximum and minimum values entered are displayed.

```
int min=0, max=0, value=0, count=0;

count = Input ("How many values will be provided? ");

value = Input ("First value: ");
min = value; max=value;          // initialise both min and max
```

```
for (int i=1; i<count; i++)  // loop for remaining items
{
   value = Input ("Next value: ");
     // determine possible new min and max
   if (min > value)
      min = value;
   if (max < value)
      max = value;
}
Output ("The min and max values are "+min+" and "+max);
```

[3 marks]

8. Write a program segment that asks the user for a sentence. The program then counts the occurrences of the characters 'u' and 'e' (the number times each appears in the sentence). The program displays the result along with the original sentence.
(*Hint: Use a loop to examine each character in the input string; useful String methods:* **.charAt(), .length()***)*

```
String sent = "";
int num_e=0, num_u=0;

sent = Input ("Please enter a sentence:");
sent = sent.toLowerCase();        // to ensure no chars are overlooked

for (int i=0; i<sent.length(); i++)
{
   if ( sent.charAt(i) == 'e' )   // if the character is a e
      num_e++;                       // incr. count
   if ( sent.charAt(i) == 'u' )   // if the character is an u
      num_u++;                       // incr. count
}

Output ("In the sentence, \""+sent+"\", there are "+
    num_e + " e's and "+
    num_u + " u's.");
```

[4 marks]

9. The Flesch Readability Index analyzes the number of syllables, words, and sentences in a document and produces a readability index number, usually between 0 and 100 indicating how difficult the text is to read (a lower index implies higher difficulty: a comic book ~ an index of 95; the New York Times ~ 39).

   Write the program segment that accepts the readability index value from the user, and outputs the required educational level to read the text. Use the table below for the ranges:

index value	educational level needed
65 - 100	primary school
50 - 64	high school
30 - 49	university student
0 - 29	university graduate

   Continue asking the user until the user decides to quit

```
String continue = "yes";   // continue input (yes to cont; otherwise stop)
int    ri=0;               // input readability index
String outstr = "";        // output message string
```

```
Output ("The readability index educational level estimator.");
do
{
   ri = Input ("What is the readability index value? ");

   if ( (65 <= ri) && (ri <= 100) )
      outstr = "primary school";
   else if ( (50 <= ri) && (ri <= 64) )
      outstr = "high school";
   else if ( (30 <= ri) && (ri <= 49) )
      outstr = "university student";
   else if ( (0 <= ri) && (ri <= 29) )        // or just: else   <-- default!
      outstr = "university graduate";

   Output ("The estimated educational level is: "+outstr);

   cont = Input ("Do you wish to estimate another? (yes/no)");
}
while ( cont.equals ("yes") );

Output ("Thank you and have a great day.");
```