Esc101N, Jan-Apr 2005, First Mid-Semester Exam Solutions
Tue 01 Feb 2005, Duration: 1 hour (14:00-15:00), Total marks: 150

Name: _____ Roll No: _____ Section: _____

Note: This exam is open hand-written (your own) notes. No xerox copies, books, other printed material, or electronic material is allowed. Please answer in the space provided for each question. You may refer to the brief notes given in the last page.

Question 00: Who is watching you? [0 marks]
Write a Java class called Mirror, and instantiate a mirror object before you attempt the questions below. The mirror object should scream if and when you attempt to copy.

Question 01: Complex, but simple [35 marks]
The following class definition is for representing complex numbers in Java, and for doing various operations. Complete the missing parts of the code.

```
class Complex {

     // Object variables [4 marks]

     double real;

     double im;

     // The constructor [6 marks]

     public Complex(double x, double y) {

          real = x;

          im = y;

     } // End constructor

     // Continued in the next page
```

```
// Method to add the given complex number to self
// [5 marks]
public void add(Complex c1) {

     real += c1.real;

     im += c1.im;

} // End add method

// Method to check if the given complex number is
// the complex conjugate of self [5 marks]
public boolean isConjugate(Complex c1) {

     return (real == c1.real) && (im == -c1.im)

} // End isConjugate method

// Write a method to copy values from a given
// complex number [5 marks]
public void copyFrom(Complex c1) {

     real = c1.real;

     im = c1.im;

} // End copyFrom method

// In the main function, instantiate two complex
// numbers c1 = 1+2i, and c2 = 20+5i.  Instantiate
// a third complex number c3 = 0+0i.  Write a series
// of two statements, at the end of which c1 and c2
// will retain their original values, and c3 will be
// c1 + c2 (21+7i).  Use only the above methods.
// [10 marks]
public static void main(String[] args) {

     Complex c1 = new Complex(1,2);

     Complex c2 = new Complex(20,5);

     Complex c3 = new Complex(0,0);

     c3.copyFrom(c1);

     c3.add(c2);
```

```
    } // End main method
} // End class definition
```

Question 02: A complex string [2 x 10 = 20 marks]

We are now going to add a class method to the complex number class definition above. The method takes a string of the form "a+bi" as argument, and creates a new complex number corresponding to a+bi. For instance, given the string "5.6+8.3i", it will return a new complex number object corresponding to 5.6+8.3i. Fill in the blanks below.

```
public static Complex parseStr(String s1) {

     int plusIndex = s1.indexOf("+");

     int iIndex = s1.indexOf("i");

     String realStr =
          s1.substring(0, plusIndex);

     String imagStr =
          s1.substring(plusIndex + 1 , iIndex ) ;

     double real = Double.parseDouble(realStr);
     double imag = Double.parseDouble(imagStr);

     return new Complex(real,imag);

} // End parseStr method
```

Question 03: Handling an alternate format [3 x 5 = 15 marks]

We are now going to modify the parseStr method above to handle "a+bi" as well as "a+ib". For instance, both "5+3i" and "5+i3" should be handled. We will consider only these two formats, and no other possibilites. We are just going to change one statement in the parseStr method above. Fill in the blanks below.

```
     // everything above in the method as earlier
     String imagStr =

          (plusIndex + 1 == iIndex) ?

          s1.substring( iIndex + 1 , s1.length());
```

```
            s1.substring( plusIndex + 1 , iIndex );
        // everything below in the method as earlier
```

Question 04: Perfect numbers [3 x 8 = 24 marks]

A perfect number is one whose factors except itself add up to that number. For instance, the factors of 6 (except 6 itself, but including 1) are 1, 2, and 3. And 1+2+3=6. Hence 6 is a perfect number. Similarly, the factors of 28 are 1, 2, 4, 7, and 14. And 1+2+4+7+14=28. Hence 28 is also a perfect number. The factors of 8 are 1, 2, and 4, but 1+2+4=7. Hence 8 is not a perfect number. Below is a method which takes a number and checks if it is perfect. Fill in the blanks.

```
public static boolean isPerfect(int N) {

      int sumOfFactors = 1 ;

      for(int i = 2; i<N ; i++ ) {

            if( (N % i) == 0 ) {

                  sumOfFactors += i;

            }

      } // End for(i)

      return  (sumOfFactors == N);

} // End isPerfect method
```

Question 05: An alternate loop [3 x 2 = 6 marks]

We are now going to consider a slight variation of the for loop above, for the same functionality. Fill in the new blanks below (there are some blanks from the previous question too – assume that these are filled the same as in the previous question).

```
      for(int i = ____; ______; ______) { // same as earlier

            if( (N % i) != 0 ) { // fill in here

                  continue; // fill in here
            }

            _________________ += ________; // same as earlier
```

```
    } // End for(i)
```

Question 06: A binary string [3 x 12 = 36 marks]

In this problem, we are going to write a method which converts a String representation of an binary integer, into the corresponding integer value. For instance, given the string "111", we get the integer 7. Given "1010", we would get 10, and so on. Fill in the blanks below.

```
public static int binStrToInt(String s1) {

     int powOf2 = 1;

     int len = s1.length();

     int number = 0;

     for(int i = len-1; i>=0 ; i-- ) {

          String digitStr = s1.substring( i , i+1 );

          int digitVal = digitStr.equals("0") ? 0 : 1 ;

          number = number + digitVal*powOf2;

          powOf2 = powOf2 * 2;

     } // End for(i)
     return number;
} // End binStrToInt method
```

Question 07: Error zone [14 marks]

The following method tries to count the number of occurrences of a given character in a given string. Point out all the errors in the code below (use the space in the next page if necessary).

```
1: public static int charCount(String s1; char c1) {
2:   int count = 0;
3:   for(i = 0; i <= s1.length; i++) {
4:        if(charAt(i) = c1) {
5:             count += count + 1; continue;
6:        }
7:   } // End for(i)
8: } // End charCount method
```

Error-1: (line 1) Function arguments should be seperated by ',' but not with ';'

Error-2: (line 3) '*i*' must be declared.

Error-3: (line 3) '*i*<=' must be '*i*<'

Error-4: (line 3) *length* must be *length()*

Error-5: (line 4) *charAt* shouldbe *s1.charAt*

Error-6: (line 4) *'= c1'* should be *'== c1'*

Error-7: (line 5) '+=' should be '='

Any more errors: missing return statement.

Question 08: Are you serious? [0 marks]

Why does the question numbering start from 0 instead of 1?

a) This does not matter since question 0 does not carry any marks
b) This course uses Java, and Java starts indexing from 0, not 1
c) There is no point in writing an exam without telling yourself that you will not copy
d) All of the above

Some useful notes:

- In case you have forgotten, the conjugate of a complex number a+bi is a–bi
- If s1 is a String, say "abcbc"
  - s1.charAt(2) is the character 'c', s1.charAt(3) is the character 'b'
  - s1.equals("abcbc") is true, and s1.equals("cba") is false
  - s1.indexOf("b") is 1, s1.indexOf("bc") is 1, and s1.indexOf("d") is –1
  - s1.indexOf("b", 0) is 1, s1.indexOf("b", 2) is 3
  - s1.lastIndexOf("b") is 3, s1.lastIndexOf("bc") is 3
  - s1.lastIndexOf("b", 2) is 1, s1.lastIndexOf("b", 3) is 3
  - s1.length() is 5
  - s1.substring(2) is "cbc", s1.substring(2, 4) is "cb"