

**THE UNIVERSITY OF WESTERN ONTARIO  
LONDON CANADA**

**COMPUTER SCIENCE 026a  
FINAL EXAMINATION  
DECEMBER 16, 2005  
3 HOURS**

NAME: \_\_ Marking Scheme \_\_\_\_\_

STUDENT NUMBER: \_\_\_\_\_

Question

1-70. \_\_\_\_\_

71. \_\_\_\_\_

72. \_\_\_\_\_

73. \_\_\_\_\_

74. \_\_\_\_\_

75. \_\_\_\_\_

TOTAL \_\_\_\_\_

(Out of 150 marks)

Important Notes:

- Be sure to fill in your name and student number on this question paper, your Scantron form, and your exam booklet. Be sure to also put course information and encode your student number in the bubbles in the appropriate section on the Scantron form.
- Circle your answers to multiple choice questions on this question paper and mark your answers on the Scantron form. Write all answers to written questions in the exam booklet provided.
- You must complete the Scantron form with a pencil and completely fill in the selected bubbles which represent your choices.
- You must submit the Scantron form, exam booklet, and this question paper.
- There are no cheat sheets, books, or other reference materials allowed for this exam. No calculators, cell phones, or other electronic devices are permitted either.

For your reference, the following information from the Java String API is supplied:

### **charAt**

```
public char charAt(int index)
```

Returns the `char` value at the specified index. An index ranges from 0 to `length() - 1`. The first `char` value of the sequence is at index 0, the next at index 1, and so on, as for array indexing.

### **equals**

```
public boolean equals(String anotherString)
```

Compares this string to the specified object. The result is `true` if and only if the argument is a `String` object that represents the same sequence of characters as this object.

### **indexOf**

```
public int indexOf(String str)
```

Returns the index within this string of the first occurrence of the specified substring.

### **indexOf**

```
public int indexOf(String str,  
                  int fromIndex)
```

Returns the index within this string of the first occurrence of the specified substring, starting at the specified index.

### **length**

```
public int length()
```

Returns the length of this string. The length is equal to the number of characters in the string.

### **substring**

```
public String substring(int beginIndex)
```

Returns a new string that is a substring of this string. The substring begins with the character at the specified index and extends to the end of this string.

### **substring**

```
public String substring(int beginIndex,  
                        int endIndex)
```

Returns a new string that is a substring of this string. The substring begins at the specified `beginIndex` and extends to the character at index `endIndex - 1`. Thus the length of the substring is `endIndex - beginIndex`.

Part I – Multiple Choice, True/False – Choose the best answer from the choices given. Circle your answer on the paper, and fill in the answer on the Scantron form. [70 marks total, 1 mark each]

1. Computer science is the study of computers.
  - a. True
  - b. False
2. A computer is a device that executes and follows a set of instructions to carry out some kind of computing activity.
  - a. True
  - b. False
3. When we say that main memory on a computer is volatile, we mean:
  - a. That it is inherently unreliable and untrustworthy
  - b. That it frequently performs poorly under certain circumstances
  - c. That information it stores is not retained when power is discontinued
  - d. All of the above
  - e. None of the above
4. An algorithm does not have which of the following characteristics:
  - a. Ordered
  - b. Effective
  - c. Unambiguous
  - d. Terminates
  - e. None of the above; an algorithm possesses all of these characteristics
5. It is possible to program a computer to solve any problem imaginable.
  - a. True
  - b. False
6. Which of the following can be directly executed by the CPU of a computer?
  - a. Pseudo-code
  - b. High-level language code
  - c. Assembly language code
  - d. All of the above
  - e. None of the above
7. With the advent of high-level languages like Java, no one writes programs in assembly language any more.
  - a. True
  - b. False

8. The von Neumann architecture discussed in class refers to a computer design model that uses a single storage structure to hold both programs and data.

a. True  
b. False

9. Suppose that an entirely new kind of computer processor was created. Which languages would likely need to be introduced for this new processor?

a. A new machine language  
b. A new assembly language  
c. A new high-level language  
d. All of the above

e. Only a) and b)

10. Java source code is compiled into:

a. Machine code  
b. Assembly code  
c. Pseudo-code

d. Bytecode

e. None of the above

11. Which of the following types of algorithmic constructs manage the flow of control in an algorithm:

a. Sequential execution  
b. Conditional execution  
c. Repetitions

d. All of the above

e. None of the above

12. A for loop is what kind of loop:

a. Pre-tested, definite

b. Pre-tested, indefinite

c. Post-tested, definite

d. Post-tested, indefinite

e. None of the above

13. A while loop is what kind of loop:

a. Pre-tested, definite

b. Pre-tested, indefinite

c. Post-tested, definite

d. Post-tested, indefinite

e. None of the above

14. A do-while loop is what kind of loop:
- a. Pre-tested, definite
  - b. Pre-tested, indefinite
  - c. Post-tested, definite
  - d. Post-tested, indefinite
  - e. None of the above
15. Which of the following is a concern when writing algorithms:
- a. Space efficiency
  - b. Time efficiency
  - c. Correctness
  - d. All of the above
  - e. None of the above
16. It should be possible to translate a well-written algorithm into any programming language.
- a. True
  - b. False
17. Which of the following is not a primitive data type in Java:
- a. int
  - b. double
  - c. char
  - d. String
  - e. None of the above; all of the above are, in fact, primitive data types
18. In Java, the curly bracket } is used to terminate:
- a. The end of a single statement
  - b. The end of a statement block or compound statement
  - c. The end of a comment
  - d. The end of a string constant
  - e. None of the above
19. Suppose you wanted to store the integer value 10 billion into a variable in Java. In this case, the variable should be of type:
- a. long
  - b. int
  - c. short
  - d. byte
  - e. It is not possible to store that large an integer value in Java

20. Suppose you wanted to store integer values between 1 and 100. The most space efficient way in Java to do so is to use a variable of type:

- a. long
- b. int
- c. short
- d. byte
- e. There is no difference between any of the above options

21. The Java expression  $2 + 2 / 5$  evaluates to:

- a. 0
- b. 0.8
- c. 2
- d. 2.4
- e. None of the above

22. The Java expression  $2.0 + 2 / 5$  evaluates to:

- a. 0.0
- b. 0.8
- c. 2.0
- d. 2.4
- e. None of the above

23. According to Java conventions, the identifier `MAXIMUM_VALUE` is most likely:

- a. A variable
- b. A method
- c. A constant
- d. A class
- e. None of the above

24. Consider the following Java code segment from the course notes:

```
int count;
double dimes;
dimes = 5;
count = dimes + 6;
dimes = count + 6;
count = (int) dimes + 6;
dimes = (double) count / 2;
```

How many of the assignment statements in this code segment are not valid?

- a. 1
- b. 2
- c. 3
- d. 4
- e. 5

25. Suppose you have the following Java code:

```
counter = counter + 1;
```

This code can be abbreviated by re-writing it as:

- a. `counter+;`
- b. `counter++;`
- c. `=counter++;`
- d. `counter + 1;`
- e. None of the above

26. Which of the following are not valid relational operators in Java:

- a. `>`
- b. `<=`
- c. `!=`
- d. `=`
- e. None of the above; all of these are valid relational operators

27. To overcome the dangling else problem, Java follows which of the following as a rule:

- a. An `else` is associated with the nearest `if`
- b. An `else` is associated with the nearest `if` that does not have a corresponding `else`
- c. An `else` that is not explicitly associated with an `if` statement will generate a compile error to the programmer
- d. An `else` that is not explicitly associated with an `if` statement will generate an exception at run-time
- e. Java has no rule to overcome the dangling else problem

28. Suppose you have a Java program that displays a menu with options numbered 1 through 5. The user's choice from the menu is input into an `int` variable called `choice`. Which of the following relational expressions will check to see whether the user's choice is outside of the correct range of values for the menu?

- a. `((count < 1) || (count > 5))`
- b. `(!(count >= 1) || !(count <=5))`
- c. `(!((count >=1) && (count <=5)))`
- d. All of the above
- e. None of the above

29. Suppose you had the following for loop in a Java program:

```
String message = "abcdefghijklk";

for (int i=message.length()-1; i >= 0; i = i - ((i % 2)+1)) {
    System.out.print(message.substring(i,i+1));
}
System.out.println();
```

This loop would produce which of the following outputs:

- a. kjihgfedcba
- b. kigeca
- c. kjhggedba
- d. kjhfdb
- e. None of the above

30. Suppose you had a while loop in a Java program that began with:

```
while(true)
```

This loop would:

- a. Never execute
- b. Execute one time only
- c. Likely loop indefinitely
- d. Generate an exception
- e. None of the above

31. Suppose you had the following do-while loop in a Java program:

```
String stars = "*****";
int count=0;
do {
    stars = stars.substring(1);
    count++;
} while (!stars.equals(""));
```

After this loop is completed, the variable `count` will contain what value?

- a. 0
- b. 1
- c. 2
- d. 3
- e. None of the above

32. Using the `break` or `continue` statements in loops in Java is generally considered to be an excellent programming practice.

- a. True
- b. False

33. Every Java program must contain at least one static method.

- a. True
- b. False

34. When you have an overloaded method in Java, it decides the particular version of a method to call based on:

- a. The number of parameters
- b. The types of parameters
- c. The return type of the method
- d. The number and types of parameters
- e. The number and types of parameters and the return type of the method

35. Suppose you have the following method prototype:

```
public static void alpha(int beta, int gamma)
```

and this method is invoked as follows:

```
int kappa=0;  
int omega=1;  
alpha(kappa, omega);
```

In this case:

- a. beta, gamma, kappa and omega are all actual parameters
- b. beta, gamma, kappa and omega are all formal parameters
- c. beta and gamma are actual parameters while kappa and omega are formal parameters
- d. beta and gamma are formal parameters while kappa and omega are actual parameters
- e. None of the above

36. A constructor:

- a. Is usually used to initialize the instance variables of an object
- b. Is invoked with the help of the keyword new
- c. Has the same name as the class
- d. All of the above
- e. None of the above

37. Using the same identifier (name) for multiple methods defined in the same class is known as:

- a. Abstraction
- b. Overriding
- c. Overloading
- d. A bad programming technique
- e. None of the above

38. An accessor method:

- a. Provides a way of modifying one or more attributes of an object
- b. Is used to create objects
- c. Is used to obtain the value(s) of one or more attributes of an object
- d. All of the above
- e. None of the above

39. A mutator method:

- a. Provides a way of modifying one or more attributes of an object
- b. Is used to create objects
- c. Is used to obtain the value(s) of one or more attributes of an object
- d. All of the above
- e. None of the above

40. A class method, as opposed to an instance method:

- a. Is created by invoking the constructor of the class
- b. Performs operations on an object
- c. Does not require an object to be invoked
- d. All of the above
- e. None of the above

41. A helper method:

- a. Should always be declared private
- b. Should always be declared public
- c. Is the main method of a program
- d. All of the above
- e. None of the above

42. Consider the following code:

```
public class BankAccount {  
    private int balance;           // line 1  
    private String accountId;     // line 2  
  
    public BankAccount(String accountId, int balance) { // line 3  
        this.accountId = accountId; // line 4  
        this.balance = balance;     // line 5  
    }  
}
```

Which one of the following statements is true?

- a. The identifier `balance` on line 3 refers to the identifier `balance` on line 1
- b. The identifier `this.balance` on line 5 refers to the identifier `balance` on line 3
- c. The identifier `this.balance` on line 5 refers to the identifier `balance` on line 1
- d. The identifier `balance` on line 5 refers to the identifier `balance` on line 1
- e. None of the above

43. Consider the following class:

```
public class Temperature {  
  
    public static void main(String[] args) {  
        int temperature = 1 ;  
        System.out.println(adjust(temperature, false)) ;  
        System.out.println(isFreezing(temperature)) ;  
    }  
  
    public _____ int adjust(int temp, boolean isCold) { // line 7  
        if (isCold)  
            temp--;  
        else  
            temp++;  
        _____ ; // line 12  
    }  
  
    public static _____ isFreezing(int temp) { // line 14  
        return temp <= 0 ;  
    }  
}
```

Which of the following keywords can be used to complete the missing parts of line 7, 12, and 14, respectively?

- a. void, return true, int
- b. static, return isCold, double
- c. static, return temp, boolean
- d. static, return temp, int
- e. void, return false, int

44. What is the output of the program in Question 43, after completing it?

- a. 1true
- b. 2false
- c. 0true
- d. 0false
- e. None of the above

45. Consider the following two strings:

```
String s1 = new String("Jack went up the hill") ;  
String s2 = new String("Jack went up the hill") ;
```

and the following expressions:

1. `s1 == s2`
2. `s1 != s2`
3. `s1.equals(s2)`
4. `!s1.equals(s2)`

Which of the above expressions evaluate to true?

- a. 1 and 3
- b. 2 and 3
- c. 1 and 4
- d. 2 and 4
- e. None of the above

46. Consider the following code segment:

```
public class findout {  
    public static void main(String[] args) throws Exception {  
  
        String s = "Java is a lot of trouble" , rs = "";  
  
        for (int i = s.length() ; i <= 0 ; i--) {  
            rs += s.charAt(i) ;  
        }  
    }  
}
```

What does this code actually do?

- a. It copies string `s` into `rs`
- b. Nothing, as this program does not compile correctly
- c. It reverses the order of the characters in `s` and puts the result in `rs`
- d. It deletes the characters contained in `s`, and puts them in `rs`
- e. None of the above

47. What does the following code segment print, when executed?

```
String s = "Hello";
for (int i = 0 ; i < s.length(); i++)
    System.out.print(s.substring(i));
```

- a. Helloelloolloo
- b. Hello ello llo lo o
- c. H e l l o
- d. o l l e H
- e. None of the above

48. Consider the following code segment:

```
String T = "Some text in here", W = "";
for (int i = 0 ; T.charAt(i) != ' ' && i < T.length() ; i++) {
    W += T.charAt(i) ;
}
System.out.println(W) ;
```

What will be printed by this code segment?

- a. here
- b. in
- c. text
- d. Some
- e. None of the above

49. Suppose one wants to declare, construct, and initialize an array containing 6 integers. Choose a valid statement to accomplish this:

- a. `int[] arr = {0, 1, 2, 3, 4, 5};`
- b. `int[] arr = new int[6];`
- c. `int[] arr;`
- d. `arr = new int[6];`
- e. None of the above

50. Consider the following code segment:

```
int[] arr = new int[5];
for (i = 0 ; i < arr.length; i++)
    arr[i] = i;
```

Which of the following statements would result in `arr` having the same values as those assigned by the above code segment?

- a. `int[] arr = {1, 2, 3, 4, 5, 6};`
- b. `int[] arr = {0, 1, 2, 3, 4, 5};`
- c. `int[] arr = {1, 2, 3, 4, 5};`
- d. `int[] arr = {0, 1, 2, 3, 4};`
- e. None of the above

51. Consider the following Java method:

```
public static boolean what(int[] arr) {
    for (int i = 0 ; i < arr.length ; i++) // line 1
        if (arr[i] == 0) return true;
    return false;
}
```

What does the method do?

- a. It initializes each element of the array `arr` to 0
- b. It returns `true` if each element of `arr` is 0
- c. It returns `true` if at least one element of `arr` is 0
- d. It always returns `false`
- e. None of the above

52. Consider the method of Question 51. If we declared an array called `ever` in this way:

```
int[] ever = new int[5];
```

Which one of the following calls to the method `what` would be valid?

- a. `what(ever) ;`
- b. `what(ever[]) ;`
- c. `what(ever[5]) ;`
- d. All of the above
- e. None of the above

53. Consider the method of Question 51 once again, and suppose we replaced the line marked `//line 1` with the line:

```
for (int i = 0 ; i < 5 ; i++)
```

A call to the method would still be syntactically valid for an array of length:

- a. 5
- b. 6
- c. 100
- d. All of the above
- e. None of the above

54. What is the effect of the following code segment?

```
int[] arr new int[100] ;
arr[0] = 0 ;
for (int i = 1 ; i < arr.length ; i++)
    arr[i] = arr[i-1] ;
```

- a. It initializes each element of the array to 0
- b. It only sets `arr[0]` to zero
- c. The code creates a runtime error
- d. The code creates a compile time error
- e. None of the above

55. Consider the following class:

```
public class R {
    private double n ;

    public R(double n) {
        this.n = n ;
    }

    public void A(double n) {
        this.n = n ;
    }

    public double B() {
        return this.n ;
    }

    public R C(R a) {
        return new R (this.B() + a.B()) ;
    }
}
```

What is the name of the constructor method?

- a. R
- b. A
- c. B
- d. C
- e. None of the above

56. Consider the class from Question 55. What is the name of the accessor method?

- a. R
- b. A
- c. B
- d. C
- e. None of the above

57. Consider the class from Question 55. What is the name of the mutator method?

- a. R
- b. A
- c. B
- d. C
- e. None of the above

58. Suppose that the following main program uses the class of Question 55:

```
public class D {  
    public static void main(String[] args) {  
  
        R a = new R(0.0) ;  
        R b = new R(0.0) ;  
        R c ;  
  
        a.A(5.0) ;  
        b.A(-4.0) ;  
        c = a.C(b) ;  
        System.out.println(c.B()) ;  
    }  
}
```

What is the output of this program?

- a. 5.0
- b. -4.0
- c. -1.0
- d. 1.0
- e. None of the above

59. Is the constructor method from the class in Question 55 overloaded?

- a. No
- b. Yes
- c. The class does not have a constructor
- d. Constructors are always overloaded
- e. None of the above

60. What is the attribute in the class defined in Question 55?

- a. A
- b. R
- c. n
- d. The class does not have any attribute
- e. None of the above

61. List the behaviours in the class from Question 55:

- a. R, A, B, and C
- b. A, B and C
- c. C only
- d. The class has no behaviours
- e. None of the above

62. What would be the effect of changing the keyword `private` to `public` in the class defined in Question 55?

- a. It would prevent any program using the class from compiling correctly
- b. It would deny access to the class attribute
- c. It would allow a program using the class to directly access the attribute
- d. It would not change anything
- e. None of the above

63. What does method `c` from the class in Question 55 perform?

- a. It returns an object containing the sum of the attributes of two previously created objects.
- b. It creates three objects
- c. It does not perform any meaningful operation
- d. None of the above

64. What is the instance variable in the class from Question 55?

- a. `a`
- b. `this.a`
- c. `b`
- d. `this.b`
- e. `n`

65. In general, what is the purpose of the method `toString()`?

- a. It instructs Java on how to print the contents of an object
- b. It transforms its parameter into a string variable
- c. It takes a string parameter and splits it two
- d. It reverses the contents of its string parameter
- e. None of the above

66. The parameter to a main program is: `String[] args`. What is the type of `args`?

- a. It is an array
- b. It is an array of strings
- c. It is a string
- d. It is string of arrays
- e. None of the above

67. Consider the following Java code:

```
int i;  
double j = 10.0;  
i = (int) j;
```

The last line of this code is an example of:

- a. Overloading
- b. Overriding
- c. Type casting
- d. Bad programming
- e. None of the above

68. A constructor method:

- a. Must have the same name as the class it is defined in
- b. Can be overloaded
- c. Can perform initializations of instance variables
- d. Can receive parameters
- e. All of the above

69. What is the output of the following program:

```
public class toto {  
    public static void tata(int a, int b, int c) {  
        a = 10 ;  
        b = 10 ;  
        c = 10 ;  
    }  
  
    public static void main(String[] args) {  
        int i = 9, j = 8, k = 7 ;  
        tata(i,j,k) ;  
        System.out.println(i + " " + j + " " + k) ;  
    }  
}
```

- a. 9 8 7
- b. 10 10 10
- c. 1 2 3
- d. 3 2 1
- e. None of the above

70. Parameter passing in Java is always :

- a. By reference of value
- b. By reference
- c. Both by value and reference
- d. By value
- e. None of the above

Part II – Written answer questions – Write your answers to these questions in the exam booklet provided. [80 marks total]

71. Consider the following class and write an appropriately named Java program that uses it to multiply two complex numbers  $(1.0 + 4.0i)$  and  $(5.0 - 3.0i)$  and output the result to the user. No input or other interaction with the user is required; it only needs to carry out this specific computation and output the results. [15 marks]

```
public class cplx {
    private double a ; // The real component of the number
    private double b ; // The imaginary component of the number

    public cplx(double a, double b) {
        this.a = a ;
        this.b = b ;
    }

    public double getRe() {
        return this.a ;
    }

    public double getIm() {
        return this.b ;
    }

    public void setRe(double a) {
        this.a = a ;
    }

    public void setIm(double b) {
        this.b = b ;
    }

    public String toString() {

        String plusorminus ;
        plusorminus = " + " ;
        if (this.b < 0.0) {
            plusorminus = " - " ;
            b *= -1.0 ;
        }
        return "(" + this.a + plusorminus + this.b + "i" ;
    }

    public cplx cplxMult(cplx n) {
        double a, b ;

        a = this.getRe()*n.getRe() -this.getIm()*n.getIm() ;
        b = this.getRe()*n.getIm() + this.getIm()*n.getRe() ;
        return new cplx(a,b) ;
    }
}
```

One possible solution is:

```
public class ComplexTest {
    public static void main(String [] args) {

        cplx[] n = new cplx[3] ;

        n[0] = new cplx(1.0,4.0) ;
        n[1] = new cplx(5.0,-3.0) ;
        n[2] = n[0].cplxMult(n[1]) ;

        System.out.println(n[2]) ;
    }
}
```

Variations include:

- Instead of using an array, declaring individual objects.
- Not using a third object, but updating one of the old reference variables.
- Not using println directly, but calling toString to do things explicitly.

72. Design and write a publicly visible static Java method to determine whether a given integer is a prime number or not. You must do so according to the following specifications. [15 marks]
- Remember that a prime number, by definition, is a number divisible only by itself and by 1.
  - The method must not perform any keyboard input, screen output, or file I/O (input/output); that is, any required values must be passed as parameters.
  - A `boolean` value must be returned, reflecting whether the given integer is passed into the method a prime number or not.
  - The method must be named appropriately.
  - Since testing a large number to determine if it is prime can be computationally expensive, you should do your best to reduce the number of calculations required in the process.

One possible solution is:

```
public static boolean isPrime(int number) {

    boolean is_prime = true;
    int count = 2;

    if (number > 0) {
        while ((is_prime) && (count < number)) {
            if ((number % count) == 0) {
                is_prime = false;
            }
            count++;
        }
    } else {
        is_prime = false;
    }

    return is_prime;
}
```

73. Write a Java code segment that gets 10 floating point values from the user, and counts the number of these values that are also integral values (i.e. they have no fractional component, as in 42.0). Your code should satisfy the following requirements.  
[15 marks]

- a. Avoid repeated code by using appropriate programming structures.
- b. The user must be prompted with a message before each input.
- c. Each number entered by the user must be read into a `double` variable. You do not need to do exception handling to ensure that the input from the user is a floating point number; you may safely assume so for this question.
- d. The code segment should be part of the given main method. You can assume the existence of the code below. (You need not copy the given code into your answer.)

```
public static void main(String[] args) throws Exception {  
    BufferedReader keyboard = new BufferedReader(  
        new InputStreamReader(System.in), 1);
```

A complete Java program meeting these specifications is:

```
import java.io.*;  
  
public class IntCheck {  
    public static void main(String[] args) throws Exception {  
  
        BufferedReader keyboard = new BufferedReader(  
            new InputStreamReader(System.in), 1);  
  
        final int NUM_NUMBERS = 10;  
        int int_count = 0;  
        double floating_value;  
  
        for (int count=0; count < NUM_NUMBERS; count++) {  
            System.out.print("Enter floating point value #" + (count + 1)  
                + ": ");  
            floating_value = Double.parseDouble(keyboard.readLine());  
            if (floating_value == (int) floating_value) {  
                int_count++;  
            }  
        }  
        System.out.println(int_count + " value(s) were integral values");  
    }  
}
```

Note that it is possible to do this using strings as well, but it gets messy quick.

74. Write a publicly visible static Java method `isPalindrome(String aWord)` that returns `true` if the contents of `aWord` as a whole is a palindrome, and returns `false` otherwise. [15 marks]

One possible solution is:

```
public static boolean isPalindrome(String aWord) {
    for (int i = 0 ; i < aWord.length()/2 ; i++) {
        if (aWord.charAt(i) != aWord.charAt(aWord.length() - i - 1)) {
            return false ;
        }
    }
    return true ;
}
```

Another variation would be to reverse the string and do a state equality check on the string to see if they match.

75. Consider the class in Question 71 and answer the following questions. [20 marks total]
- a. List the constructor method(s) [5 marks]

`cplx(double a, double b)`

- b. List the accessor method(s) [5 marks]

`getRe(), getIm(), cplxMult()`

- c. List the mutator method(s) [5 marks]

`setRe(double a), setIm(double b), toString()`

- d. What is the scope of `private double a` and `private double b`? In other words, in what part of the supplied class code are these particular instance variables directly accessible? [5 marks]

`private double a` and `private double b` are accessible only within class `cplx`