

THE UNIVERSITY OF WESTERN ONTARIO

Computer Science 1026b, Spring 2010

Computer Science Fundamentals I

Final Exam Solutions

Solutions

§1 Very Short Answers

The Java code in the following questions is valid and will provide output (without error) if entered into the DrJava interactions pane.

Question 1

Each answer in this Question is worth 1 point for a total of 10 points.

- a. After the following code segment is executed

```
int x = 10 + 2*5 % 10
```

the value of x is:

10

- b. The number of "*"s printed by

```
for (int star = 9; star < 0; star++) {  
    System.out.println("*");  
}
```

is:

0 (none)

- c. Suppose we have defined

```
int a = 12, b = 0;  
boolean t = true;
```

In order to make the expression `(t ? !(a > b)) && t` evaluate to `true`, we should replace

? with:

|| (or), != (not equal)

- d. Suppose we have defined

```
int m=18, n=4;
```

then the value of `m / n + (m % n)` is:

6

e. After the following code segment is executed

```
int x = 3;  
x = x * x;  
x = x + x;
```

the value of `x` is:

18

f. Suppose we have defined

```
String s1 = new String("Finals");  
String s2 = new String("Finals");
```

then the value of the expression `(s1==s2)` is:

false

g. The *last* value of the array `int[] xs = {1,2,3,4}` is at position:

3

h. After the following code segment is executed

```
int x;  
for (x = 1; x < 5; x = x + x);
```

the value of `x` is:

8

i. After the following code segment is executed

```
int num = 15;  
if (num >= 20)    num = num - 20;  
if (num >= 10)   num = num - 10;  
if (num >= 5)    num = num - 5;
```

the value of `num` is:

0

j. After the following code segment is executed

```
System.out.println("3"+"7");
```

what is printed?

37

Question 2

Each answer in this Question is worth *2 points* for a total of *10 points*.

- a. For 32 to be printed by the following code segment

```
int total = x;
for (int i = 0; i <= 10 ; i = i + 2)
{
    total += i;
}
System.out.print(total);
```

the value of x should be:

2

- b. The shape drawn by the following Turtle class method

```
void SecretShape () {
    for (int i =0; i < 13; i++) {
        this.forward(30);
        this.turnLeft();
    }
}
```

is a:

square

- c. After the following code segment is run

```
int z = 0;
double x = 123./321, y = 0.38317757009345793;
if ( x <= y && x > y ) {
    z = 1;
}
```

the value of z is:

0

d. After the following code segment is run

```
int sum = 0;
for (int i = 0; i < 10; i ++) {
    sum += i ;
    i ++;
}
```

the value of `sum` is:

20

e. After the following code segment is run

```
int x = 2*2*2*3*3*3*7;
while ( (x % 2 == 0) || (x % 3 == 0) ) {
    if (x % 2 == 0) {
        x = x/2;
    } else {
        x = x/3;
    }
}
```

the value of `x` is:

7

§2 Short Answers

Each solution in this section is worth *4 points* for a total of *20 points*.

Question 3

Write a java class method `int MinElement (int[] xs)` that returns the *least* (or *smallest*) element of the array `xs`. For instance, if we had the array `A = {-3,4,10,-32,0}` then `MinElement(A)` would return `-32`.

```
int MinElement (int[] xs) {
    int min = xs[0];

    for (int i=0; i < xs.length; i++) {
        if (xs[i] < min) min = xs[i];
    }
    return min;
}
```

Question 4

Write a java expression that is equivalent to the algebraic expression:

$$\frac{\frac{c - (a + b)}{d} \times e}{f}.$$

You may assume that the initialization

```
double a,b,c,d,e,f;
```

has been made.

```
(( ( c - (a+b) ) / d ) * e ) / f
```

Question 5

Write a for loop that, given `String s`, prints the string backwards. You may assume that the string is already assigned to the variable `s`.

For example, given `String s = "abcde"`; the for loop would print `edcba`.

```
char cs[] = s.toCharArray();
for (int i = s.length()-1; i >= 0; i--) {
    System.out.print( cs[i] );
}
```

Question 6

Write a java class method `void AnnoyUser()` that repeatedly prompts the user for input until the string "banana" is entered. (Hint: use `SimpleInput.getString` and a `while` loop).

```
void AnnoyUser() {
    String test;
    boolean done = false;
    while (! done) {
        test = SimpleInput.getString()
        if (test.equals("banana")) {
            done = true;
        }
    }
}
```


Question 7

There are four syntax errors in the following code segment:

```
/00/ class Foo {
/01/     public void FooBar ( ) {
/02/         int xs = {1,2,3,4};
/03/
/04/         for ( i = 0; i <= 4; i+=2 ) {
/05/             System.out.println( "The value of i is", i );
/06/         }
/07/
/08/         return  xs[0];
/09/     }
/10/ }
```

For each error, indicate the line where the error is and correct the error (for instance you could write something like: "Line 15 should be `double x = 2.5`").

Line 02 should be : `int xs[] = {1,2,3,4};`

Line 03 should be : `int i;`

-or alternatively-

Line 04 should be : `for (int i=0; i<=4; i+=2) {`

Line 05 should be : `System.out.println("The value of i is"+i);`

Line 01 should be : `public int FooBar () {`

-or alternatively-

Line 08 should be : `*empty*`

§3 Long Answers

Each solution in this section is worth *10 points* for a total of *40 points*.

Question 8

- a. Write an algorithm in pseudo-code (i.e. code mixed with English) which prints the following pattern when given n . Your method should work for any value of n and not only on the examples given below. Note that for a given n , the printed pattern has n rows and n columns.

$n = 3$	$n = 4$	$n = 5$
..1	...11
.22	..22	...22
333	.333	..333
	4444	.4444
		55555

```
for (i = 1; i<=n; i++) {
    Print (n-i) many "." 's
    Print i many "number" i
    Print a new line
}
```

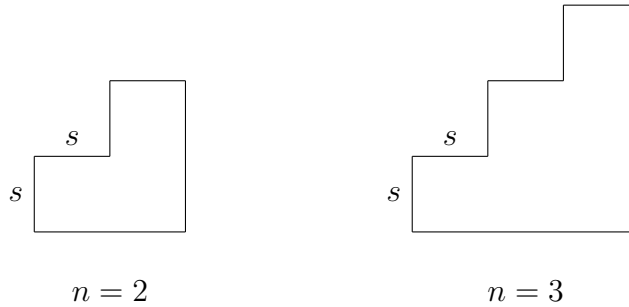
- b. Write a java class method `static void PrintPattern (int n)` that prints the same pattern as in Part a.

```
static void PrintPattern (int n) {
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n-i; j++) {
            System.out.print(".");
        }
        for (int j = 1; j <= i; j++) {
            System.out.print(i);
        }
        System.out.println();
    }
}
```

Question 9

Write a java object method `void StairCase (int n, int s)` for inclusion in the `Turtle` class that takes as input n and s and draws a staircase with n stairs of side length s . This illustrated for $n = 2$ and $n = 3$ below (but remember your code should work for *any* value of n).

You may assume that drawing these stairs will never cause your turtle to go out of its world's boundaries.



```
public void StairCase (int n, int s) {
    for (int i = 0; i < n; i++) {
        this.forward(s);
        this.turnRight();
        this.forward(s);
        this.turnLeft();
    }
    this.backwards(s*n);
    this.turnRight();
    this.backwards(s*n);
    this.hide();
}
```

Question 10

Write a java class method `int sum (int n)` that finds the sum of the first n positive integers (i.e. $1 + 2 + 3 + \dots + n$.) For example `sum(3)` returns 6 because $1 + 2 + 3 = 6$ and `sum(8)` returns 36 because $1 + 2 + 3 + \dots + 8 = 36$. Your code should work for these examples *and* any other positive integer n .

```
int sum (int n) {
    int sum = 0;
    for (int i = 1; i<=n; i++) {
        sum+=i;
    }
    return sum;
}
```

--alternate solution--

```
int sum (int n) {
    return n*(n+1)/2;
}
```

Question 11

Determining if a `Pixel` object is white is relatively straightforward: if a `Pixel` object's red, green, and blue components are all 255, then the `Pixel` object is white.

- a. Write a java method: `boolean isWhitePixel (Pixel P)`, for inclusion in the `Picture` class that tests if a pixel `P` is white.

```
boolean isWhitePixel (Pixel P) {
    if (P.getBlue()+P.getRed()+P.getBlue() < 3*255) {
        return false;
    } else {
        return true;
    }
}
```

- b. Using your method from Part a., write a java object method: `int countWhitePixels()`, for inclusion in the `Picture` class, that counts the number of white pixels in a picture. This method should take no parameters and return an integer that is the number of white pixels in the picture.

```
int countWhitePixels() {
    Pixel[] PixArr = this.getPixels();
    int count = 0;

    for (int i=0; i < PixArr.length; i++) {
        if ( PixArr[i].isWhitePixel ) {
            count++;
        }
    }

    return count;
}
```

- c. Suppose `Picture Sunset = new Picture(...)` has been properly initialized. What would you type in the interactions pane to determine the number of white pixels in `Sunset` (you may assume all classes have loaded properly).

```
Sunset.countWhitePixels();
```

§4 Very Long Answer

There is one question in this section with five parts. Each part is worth 4 points for a total of 20 points.

Question 12

For this question you will create a new class that represents information about students in a university course. Use the following code as a starting point:

```
public class Course {
    private String instructor; //the name of the instructor
    private String course; //the name of the course
    private Student[] students; //an array for the students in the course
}
```

Specifications for Student class is part of the extra exam material (i.e. the “cheat sheet”).

Part i.

Write a constructor method for inclusion in the Course class with the header:

```
public Course (String instructorName, String courseName, int maxClassSize)
```

that initializes the course in the obvious way (maxClassSize is used to initialize the students array).

```
public Course (String instructorName, String courseName, int maxClassSize) {
    instructor = instructorName;
    course = courseName;
    students = new Students[ maxClassSize ];
}
```

Part ii.

Write three methods for inclusion in the `Course` class with headers:

1. `public String getInstructor()`
2. `public String getCourse()`
3. `public int getMaxClassSize()`

which return (respectively) the name of the instructor, the name of the course, and the maximum class size.

```
public String getInstructor() {  
    return instructor;  
}
```

```
public String getCourse() {  
    return course;  
}
```

```
public int getMaxClassSize() {  
    return students.length;  
}
```


Part iii.

Write a method for inclusion in the `Course` class with the header

```
public boolean enrollStudent(Student theStudent)
```

that *adds* a students to the course. More precisely, this method returns `false` if there is no room for the student in the course; otherwise it puts `theStudent` in the `students` array and returns `true`.

```
public boolean enrollStudent (Student theStudent) {
    int i = 0;

    while (students[i] != null) {
        i++;
        if (i == students.length) return false;
    }

    students[i] = theStudent;
    return true;
}
```

Part iv.

Write a method for inclusion in the `Course` class with the header

```
public boolean removeStudent(Student theStudent)
```

that *removes* a student from the course. Return `true` if successful and `false` otherwise. (Removing a student that is not in the course should be considered a success).

```
public boolean removeStudent (Student theStudent) {

    if (students == null) return false;

    for (int i=0; i<students.length; i++) {
        if (this.equals(theStudent)) {
            students[i] = null;
            return true;
        }
    }

    return false;
}
```

Part v.

Write a method for inclusion in the `Course` class with the header

```
public int getNumberOfEnrolledStudents()
```

that returns the number of enrolled students in the course.

```
public int getNumberOfEnrolledStudents() {  
  
    int count=0;  
  
    for (int i=0; i<students.length; i++) {  
        if (students[i] != null) {  
            count++;  
        }  
    }  
  
    return count;  
}
```