

SAMPLE EXAM
First Exam
Computer Programming 326
Dr. St. John
Lehman College
City University of New York
Thursday, 7 October 2010

NAME (Printed) _____
NAME (Signed) _____
E-mail _____

Exam Rules

- Show all your work. Your grade will be based on the work shown.
- The exam is closed book and closed notes.
- When taking the exam, you may have with you pens or pencils, and an 8 1/2" x 11" piece of paper filled with notes, programs, etc.
- You may not use a computer or calculator.
- All books and bags must be left at the front of the classroom during this exam.
- **Do not open this exams until instructed to do so.**

Question 1	
Question 2	
Question 3	
Question 4	
Question 5	
Question 6	
Question 7	
Question 8	
Question 9	
Question 10	
TOTAL	

1. True or False:

- (a) ___ Pseudocode is a combination of English and a programming language.
- (b) ___ A variable can only hold numbers.
- (c) ___ The **else** part is always needed on an **if** statement.
- (d) ___ All variables in Java are global.
- (e) ___ Parameters to methods must be of primitive type.
- (f) ___ Methods with return type **void** must return an object.
- (g) ___ The method *size()* returns the number of elements in an array.
- (h) ___ Instance variables should be private.
- (i) ___ You may only have one constructor per class.
- (j) ___ Each primitive type also has a wrapper class.

2. Write the Java code that declares

- (a) a double **taxRate** that holds the number 8.725:

- (b) a character **next** which is **z**:

- (c) a string **palindrome** that holds “a man a plan a canal panama”:

- (d) an object **contentPane** of the class **Container**:

- (e) an array **pixels** of 10 **Pixel** objects:

3. What is the output of the following code fragments:

(a)

```
int numtimes = -1;
while ( numtimes <= 0 )
{
    System.out.print("Hi!");
    numtimes++;
}
System.out.print("Bye!");
```

Output:

(b)

```
boolean done = false;
int total = 1;
while ( !done )
{
    if ( total > 4 )
    {
        done = true;
    }
    total = total*2;
}
System.out.println(total);
```

Output:

(c)

```
int i, j;
for ( i = 0 ; i < 3 ; i++)
{
    for ( j = 0 ; j < i ; j++)
    {
        System.out.print("+");
    }
    System.out.println();
}
```

Output:

(d)

```
int i, j;
for ( i = 0 ; i < 6 ; i++)
{
    for ( j = 0 ; j < 3 ; j++)
    {
        if ( i%2 == 0 )
        {
            System.out.print("+");
        }
        else
        {
            System.out.print("-");
        }
    }
    System.out.println();
}
```

Output:

4. What is the output when the code is run?

(a)

Output:

```
int[] a = {1,2,3};
int[] b = {1,2,3};
if (b == a)
    System.out.println("Equal by ==.");
else
    System.out.println("Not equal by ==.");
```

(b)

Output:

```
int[] [] t = new int[10][6];
for (int row = 0; row < 10; row++)
    for (int column = 0; column < 6; column++)
        t[row][column] = getBalance(1000.00, row + 1,
            (5 + 0.5 * column));
for (int row = 0; row < 10; row++) {
    System.out.print((row + 1) + "      ");
    for (int column = 0; column < 6; column++)
        System.out.print("$" + t[row][column] + "  ");
    System.out.println( );
}
```

5. Declare and instantiate the following arrays:

(a) an array, `times`, that holds ten real numbers:

(b) an array, `count`, that holds the counting numbers from 0 to 5:

(c) a 2 dimensional array, `table`, of 5 rows by 6 columns of integers:

6. Assume the following class definition:

```
public class Mystery {  
    public int number;  
    public String message;  
    public Mystery()  
    { number = 3; message = "Hello"; }  
    public String toString()  
    { return(number+" "+message); }  
    public void query()  
    {  
        int i;  
        System.out.print(message);  
        for ( i = 0 ; i < number ; i++ )  
            System.out.print("!");  
        System.out.println();  
    }  
}
```

and the following code has been executed:

```
Mystery first = new Mystery();  
Mystery second, third;  
first.number = 2;  
first.message = "Hi";  
second = new Mystery();  
second.number = 2*first.number;  
third = first;
```

What is the output from the following statements?

(a) `System.out.print(first);`

Output:

(b) `first.query();`

Output:

(c) `System.out.print(second);`

Output:

(d) `second.query();`

Output:

(e) `System.out.print(third);`

Output:

7. Examine the class below and answer the following:

- (a) What is the output of this program?
- (b) How many times is the first `print()` method called?
- (c) How many times is the last `print()` method called?
- (d) In your own words, describe how the compiler decides which of the `print()` methods to call:

```
public class CircleCalculator {
    public static void main(String[] args) {
        double radius = 1;
        CircleCalculator.print(radius);
        double area = CircleCalculator.getArea(radius);
        CircleCalculator.print(radius, area);
        double circumference = CircleCalculator.getCircumference(radius);
        CircleCalculator.print(area, circumference);
    }
    public static double getArea(double radius) {
        return Math.PI * radius * radius;
    }
    public static double getCircumference(double radius) {
        return Math.PI * (radius + radius);
    }
    public static void print(double x) {
        System.out.println(x + " inches");
    }
    public static void print(double x, double y) {
        System.out.println(x+" inches and"+y+ " inches");
    }
}
```

8. (a) Write a method that returns the average of three integers:

```
public static int average(int first, int, middle int last) {
```

}

- (b) Write a method that takes an array of doubles and returns the smallest:

```
public static double smallest( double[] in) {
```

}

- (c) Write a method that interchanges two elements in an array of `Graphics` at the given indices

```
public static void swap( Graphics[] a, int first, int last) {
```

}

9. You have just accepted a job with a local bookstore. Your first assignment is to keep track of inventory of books at the store. Your predecessor, before quitting, began writing a `Book` class. Each of the methods of the class is preceded by a comment that explains what the method should do. Fill in each method with the appropriate code:

```
public class Book
{
    public String title;        //The title of the book
    public int numCopies;       //Number of copies of the book
    public int numRequests;     //Number of people requesting book
    public double price;        //Price of book
    public String[] requests;   //A list of customers waiting for the book

    public Book(String t, int n, double p) {
        title = t; numCopies = n; price = p; requests = new String[10];
    }
    /* Prints all the information about the book: */
    public String toString()
    {

    }

    /* Calculates and returns the number of books available (ie the difference
       between numCopies and numRequests). */
    public int numAvailable()
    {

    }

    /* Returns true if there's 1 or more books in stock, otherwise returns false*/
    public boolean inStock()
    {

    }

    /* Returns the first name on the waiting list for the book:
    public String nextInLine()
    {

    }

}
```


10. Create a new class called `GrayScale` that implements `ActionListener`. Your class has access to a global variable `picture` of the `Picture` class and `pictureFrame` of class `PictureFrame`. Your class should refresh the `pictureFrame` with a gray scale version of the picture.

```
class GrayScale implements ActionListener {
    public void actionPerformed (ActionEvent e) {
```

} }

The API for the `Picture` and `Pixel` classes are included on the next page:

Class Picture

```
java.lang.Object
└── Picture
```

```
public class Picture
    extends Object
```

The Picture class stores a two-dimensional image. This Picture is comprised of pixels (see [Pixel](#)) and is displayed on-screen through the use of a [PictureFrame](#).

Author:
Richard Wicentowski

Constructor Summary

Picture (int width, int height)	Create a Picture with the specified width and height.
Picture (Picture source)	Create a Picture by copying the contents of another (non-null) Picture.
Picture (String filename)	Create a Picture from a file.

Method Summary

BufferedImage getBufferedImage ()	Returns the BufferedImage underlying the Picture.
int getHeight ()	Return the height of the Picture.
Pixel getPixel (int x, int y)	Return the Pixel stored at this location.
int getWidth ()	Return the width of the Picture.
Pixel setPixel (int x, int y, Pixel pixel)	Set the location (x,y) to a new Pixel value.
void updateImage ()	Flush changes to the underlying image.

Class Pixel

```
java.lang.Object
└── Pixel
```

```
public class Pixel
    extends Object
```

The Pixel is the underlying data-structure in a [Picture](#). Each Pixel represents a color, stored as separate integer values for red, green and blue. Each value is between 0 and 255. Collectively, these RGB (Red-Green-Blue) values form the color of the Pixel.

Author:
Richard Wicentowski

Constructor Summary

Pixel (int r, int g, int b)	Create a Pixel with the specified RGB values.
---	---

Method Summary

boolean equals (Object other)	Determine if this Pixel is equal to another Object.
int getBlue ()	Returns the blue component of the Pixel.
int[] getComponents ()	Returns the RGB components of this Pixel as a 3-place int array.
int getGreen ()	Returns the green component of the Pixel.
int getRed ()	Returns the red component of the Pixel.
void setBlue (int blue)	Set the blue component of the Pixel.
void setGreen (int green)	Set the green component of the Pixel.
void setRed (int red)	Set the red component of the Pixel.
String toString ()	Return the String representation of this Pixel as a triple (R, G, B).