

The following problems have been used on previous exams for CS212. The exam will have 4 problems.

1. Write a function called *shorten* that is defined in the class P8 and shortens each element of an array of strings. Every string with more than two characters is cut down to its first two characters.

For example, a program that uses the function *shorten* follows.

```
public class P8 {
    public static void main(String args[]) {
        String x[] = {"CSCI", "1", "11", "Queens", "College", "CUNY"};
        shorten(x);
        for (int i = 0; i < 6; i++) System.out.print(x[i] + " ");
        // Output:  CS 1 11 Qu Co CU
        System.out.println();
    }
}
```

**Answer:**

2. Write the best **title lines** for the functions that are called by the following main program. **Do not supply the blocks for the functions.**

```
public class P1 {
    public static void main(String args[]) {
        int x = 2, y = 3, z[] = new int[4];
        boolean a = true, b = false, c[] = new boolean[4];
        String s = "Hello", t = "goodbye", u[][] = new String[4][5];
        for (int i = 0; i < 4; i++) c[i] = data(x, y, 2.5);           // (a)
        setToFive(z, c);
        System.out.println(z[1]);                                   // (b) prints 5
        System.out.println(speedLimit(x, z[1]));                  // (c) prints 55
        System.out.println(numberStrings(4, u, 5));               // (d) prints 20
        f(numberStrings(0, u, 0), data(y, x, f(20, a || b)));     // (e)
    }
}
```

- (a) Title line for **data** as called at the line marked (a).

**Answer:**

- (b) Title line for **setToFive** as called at the line marked (b).

**Answer:**

- (c) Title line for **speedLimit** as called at the line marked (c).

**Answer:**

(d) Title line for `numberStrings` as called at the line marked (d).

**Answer:**

(e) Title line for `f` as called at the line marked (e).

**Answer:**

3. Consider the following Java program.

```
public class P2 {
    public static int F(int x[], int c) {
        if (c < 3)
            return 0;
        return x[c - 1] + F(x, c - 1);
    }

    public static int G(int a, int b) {
        b = b - a;
        a = b + a;
        return a;
    }

    public static void main(String args[]) {
        int a = 4, b = 1;
        int x[] = { 3, 1, 4, 1, 5 };
        String s = "Problem Number 2";
        System.out.println(x[2 + 2] + x[2]);           // line (a)
        System.out.println(s.substring(2, 3));        // line (b)
        System.out.println(s.substring(s.indexOf("b"))); // line (c)
        System.out.println(" + G(b, a) + a + b);      // line (d)
        System.out.println(F(x, 5));                  // line (e)
    }
}
```

(a) What is the output at line (a)?

**Answer:**

(b) What is the output at line (b)?

**Answer:**

(c) What is the output at line (c)?

**Answer:**

(d) What is the output at line (d)?

**Answer:**

(e) What is the output at line (e)?

**Answer:**

4. Write blocks of code to perform the functions used in the following main program. Your blocks must match the given title lines. Each block should be a short function of only a few lines.

```
public class P3 {
    public static void main(String args[]) {
        int a[] [] = {{1, 2, 3, 4}, {0, 1, 2, 3}};
        int b[] = {3, 1, 4, 1};
        int x = 1, y = 2;
        String s = "hello";
        // (a) Return true if at least one of x and y is positive. Here Yes is printed
        if (positive(x, y)) System.out.println( "Yes" );
        // (b) Return the sum of the first row. Here 10 is printed.
        System.out.println(rowSum(a));
        // (c) Return the smallest element. Here 1 is printed.
        System.out.println(smallest(b));
        // (d) Remove the first letter. Here ello is printed.
        System.out.println(removeFirst(s));
        // (e) Insert an X at the specified position. Here heXllo is printed.
        s = addX(s, 2);
        System.out.println(s);
    }
}
```

(a) `private static boolean positive(int x, int y)`

**Answer:**

(b) `private static int rowSum(int[] [] a)`

**Answer:**

(c) `private static int smallest(int[] b)`

**Answer:**

(d) `private static String removeFirst(String s)`

**Answer:**

(e) `private static String addX(String s, int i)`

**Answer:**

5. Write a function called *randFill* that fills the entries of an array with random integers in the range from 10 to 99 (inclusive). (You should use a standard Java method to generate the values. Your solution should use no more than 6 lines of code.)

For example, a program that uses the function *randFill* follows.

```
public class P4 {
    public static void main(String args[]) {
        int x[];
        x = randFill(5);
        for (int i = 0; i < 5; i++)
            System.out.print(x[i] + " "); // prints 5 random numbers
        System.out.println();           // such as 93 73 12 69 40
    }
}
```

**Answer:**

6. Write a function called *evenUp* that returns the result of increasing the first even digit in a positive integer parameter by 1. (Your solution should use no more than 10 lines of code. Your function can return any convenient value of your choice if the parameter is not positive.)

For example, a program that uses the function *evenUp* follows.

```
public class P5 {
    public static void main(String args[]) {
        System.out.println(evenUp(1232)); // prints 1332 only the first even 2 changes
        System.out.println(evenUp(1332)); // prints 1333
        System.out.println(evenUp(1333)); // prints 1333 no even digit to change
        System.out.println(evenUp(22));   // prints 32
        System.out.println(evenUp(2));    // prints 3
    }
}
```

**Answer:**

7. Write a complete Java program that is to be used for a psychology study into random number choices by a human volunteer. Your program is to operate as follows. (Programs that correctly carry out some of the tasks will receive partial credit. Your program should not be more than 30 lines long.)

Ask the user (the volunteer) to repeatedly type 2 digit numbers onto the screen.

Read the user input and discard any number that is less than 10 or greater than 100, but keep track of numbers within this range.

When the user enters the number 100 the experiment ends and the program prints a summary with the following form (with one line of output for each of the numbers from 10 to 99 that was used):

User chose 10 for 2.1% of choices.

User chose 12 for 0.7% of choices.

User chose 16 for ...

**Answer:**

8. Read the following code and answer the questions starting at part(a).

```
public class Q1 {  
  
    public static void main(String[] args) {  
        print(args);  
    }  
  
    static void print(String[] greetings) {  
        for (int i = 0; i < args.length; i++)  
            System.out.println(greetings[i] + " ");  
        return true;  
    }  
}
```

- (a) Does this program compile? If yes, proceed to part (c).

**Answer:**

- (b) If your answer to part (a) is no, correct and rewrite the line(s) that has(have) error(s).

**Answer:**

- (c) What is the name of the file that must contain class Q1 so that you can compile it using the command line in the terminal?

**Answer:**

- (d) What command would you type on the terminal to run the program in class Q1 so as to produce output: "Welcome to cs212!" ?

**Answer:**

9. Write a function called *positiveRows* that returns the number of rows of a 2-dimensional array that have a positive sum. The array contains doubles.

For example, a program that uses the function *positiveRows* follows. The output is 2 because only columns 1 and 3 have positive sum.

```
public class P7 {
    public static void main(String args[]) {
        double x[][] = { { 1.0, 6.0, 3.0, -5.0, -4.0, 2.0 },
                          { -4.0, -4.0, -4.0, 4.0, 2.0, 4.0 } };
        System.out.println(positiveRows(x)); // prints 1
    }
}
```

**Answer:**

10. Read the following code. Then, in each of the labeled lines, find all syntax errors and state the nature of the errors in the space provided for the answer. If there is no error, write "No error" in the answer space.

```
class A {

    private int count;
}

class C {
    private String text = "";

    public void C(String s) {
        String text = s;
    }
}
```

```

class D {
    int x = 3;
    boolean b;

    public D(boolean b) {
        this.b = b;
    }

    public int doIt() {
        return this;           //-----(a)
    }
}

public class Y {

    public static void main(String[], args) {

        A a = new A();
        a.count = 10;         //-----(b)

        C c = new C("ABC");   //-----(c)
        System.out.println(c.text.toLowerCase()); //-----(d)

        D d = new D(true);
        System.out.println(d.doIt());
    }
}

```

(a) **Answer (a):**

(b) **Answer (b):**

(c) **Answer(c):**

(d) **Answer(d):**

11. In this problem you are to write complete code for a class called **TrafficSignal**. The class represents a traffic signal that cycles through four states: *green*, *yellow*, *red* and *leftTurn*, in this order. This means that *leftTurn* comes just after *red* and before *green*.

The class should have only one attribute, *light*, which is of type `int`. The value of this attribute determines the state of the traffic signal. For example, if the attribute *light* has value 0, the traffic signal shows *green*.

The class should have one (default) constructor and only three additional methods. The default color of the light should be *red*. The three methods are **currentLight()** which returns the integer corresponding to the current state, **changeLight()** which changes light from the current state to the next state. Finally, write a `toString()` method which returns the state of the traffic signal as *green*, *yellow* etc..

**Answer:** Write your class below.

12. Predict the output generated at the marked *println* lines in the following program. The program makes use of the *class Employee* that is also given. Please enter your answers in the space provided below the code.

```
public class Employee {

    private String name;
    private double salary;

    public Employee(String name, double salary) {
        this.name = name;
        this.salary = salary;
    }

    public String getName() {
        return name;
    }

    public double getSalary() {
        return salary;
    }

    public void raiseSalary(double percent) {
        double raise = salary * percent/100;
        salary += raise;
    }
}

public class Pr3 {
```



```

public static void main(String[] args) {
    double percent = 10;
    System.out.println("percent = " + percent);    //-----(a)
    doubleValue(percent);
    System.out.println("percent = " + percent);    //-----(b)

    Employee john = new Employee("John", 75000);
    System.out.println("Salary: $" + john.getSalary()); //-----(c)
    doubleSalary(john);
    System.out.println("Salary: $" + john.getSalary()); //-----(d)

    Employee a = new Employee("Emily", 90000);
    Employee b = new Employee("Henry", 100000);
    System.out.println("a's Name: " + a.getName()); //-----(e)
    System.out.println("b's name: " + b.getName()); //-----(f)
    swap(a, b);
    System.out.println("a's Name: " + a.getName()); //-----(g)
    System.out.println("b's name: " + b.getName()); //-----(h)
}

public static void doubleValue(double x) {
    x = 2 * x;
    System.out.println("End of method: x = " + x); //-----(i)
}

public static void doubleSalary(Employee y) {
    y.raiseSalary(200);
    System.out.println("End of method: Salary : " + y.getSalary()); //-----(j)
}

public static void swap(Employee x, Employee y) {
    Employee temp = x;
    x = y;
    y = temp;
    System.out.println("End of method: x is " + x.getName()); //-----(k)
    System.out.println("End of method: y is " + y.getName()); //-----(l)
}
}

```

(a) **Answer(a):**

(b) **Answer(b):**

(c) **Answer(c):**

(d) **Answer(d):**

(e) **Answer(b):**

(f) **Answer(e):**

(g) **Answer(f):**

(h) **Answer(g):**

(i) **Answer(h):**

(j) **Answer(i):**

(k) **Answer(j):**

(l) **Answer(k):**

(m) **Answer(l):**

13. (a) Retail products are identified by their **Universal Product Codes (UPCs)**. The most common form of a UPC code has 12 decimal digits: the first digit  $x_1$  identifies the product category, the next five digits  $x_2, \dots, x_6$  identify the manufacturer, the following five digits  $x_7, \dots, x_{11}$  identify the particular product, and the last digit  $x_{12}$  is a check digit that is determined by the following relation:

$$(3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12})\%10 = 0$$

Write a method called `checkDigit()` that takes a `String` parameter whose 11 characters are the first 11 digits of a UPC code. The method should return an `int` that gives the value of the corresponding check digit.

```
public static int checkDigit(String str) {
```

**Answer:**

- (b) Write a fast recursive method for evaluating  $a^n$  when  $n \geq 0$  is a non-negative integer and  $a$  is a positive integer greater than 1. [*Hint: use the equalities  $a^{2n} = (a^n)^2$  and  $a^{2n+1} = a \times a^{2n}$ .*]

```
public static int fastPower(int a, int n) {
```

**Answer:**

14. This question is related to Strings and Files. It is a two part question that requires code for two methods.

- (a) Write a private method called **removeDuplicates** that accepts a `String` parameter and returns a new `String` in which consecutive occurrences of a character are removed.

If the method **removeDuplicates(football)** is called, it returns `foftbal` after replacing the consecutive o's by a single o and the consecutive l's by a single l. Similarly if we pass `cookbook!!!` as a parameter to the method, it returns `cookbok!`.

**Your Code:**

**Answer:**

- (b) Write a private method called `removeDuplicatesFromFile` that accepts a `String` parameter that names a text file. It reads the file line by line and removes from each word of the file any consecutive occurrences of a character. It prints the entire modified text with words separated by spaces, and punctuation marks.

For example, let us suppose that the file has the following text:

```
The sky is a deep red!!!  
Calvin's skin is a pale green!!  
Yellow flowers are now blue!!!!  
Theee eeend.
```

When we call `removeDuplicatesFromFile("myFile.txt")` it should print:

```
The sky is a dep red!  
Calvin's skin is a pale gren!  
Yelow flowers are now blue!  
The end.
```

You can see that the duplicate letters and punctuation marks are removed. You can assume that the file exists and the words are separated by single spaces.

#### **Your Code:**

15. A school has 100 lockers and 100 students. All lockers are closed on the first day of school. As the students enter, the first student, S1, opens every locker. Then the second student, S2, begins with the second locker, denoted by L2, and closes every other locker. Student S3 begins with the third locker L3, and changes every third locker (closes if it is open, and opens if it is closed). Student S4 begins with the locker L4 and changes every fourth locker. Student S5 begins with L5 and changes changes every fifth locker, and so on, until S100 changes L100.

After all students have passed through the building and changed the lockers, some lockers will be open and others will be closed. For example, L1 was only changed by S1 and will be open. However, L3 was changed by S1 and S3 so will be closed. Write a program to find and display all lockers that will be open. Your output should list the open lockers as L followed by a number and separate different lockers by exactly one space. There should be no more than 15 actual instruction lines excluding all braces and empty lines. Name your class **Lockers**. It should have just a static main method.

(**Hint:** Use an array of 100 boolean elements, each of which indicates whether a locker is open(*true*) or closed(*false*). Initially all lockers are closed.)

16. In both parts read the code and predict the output.

```

public static double mystery(double d[], int l, int r) {
    if( l == r) return d[l];
    int m = l + (r-l)/2;
    double x = mystery(d, l, m);
    double y = mystery(d, m+1, r);
    if (x > y) return x;
    else return y;
}

```

(a) what does the above method return?

```

public static int mystery1(int b, int n) {
    if ( n == 0) return 1;
    int y = mystery1(b, n/2);
    int z = y * y;
    if (n%2 == 0)
        return z;
    else
        return z*b;
}

```

(b) What will be printed by the following instruction?

```

System.out.println(mystery1(2,10));

```

17. (a): Write a *class Distance* with the following specification: There are two instance variables: *feet* and *inches* stored as integers; a private constant called *INCHES\_IN\_A\_FOOT* with value 12; a no argument constructor to initialize a *Distance* as zero feet, zero inches.

**Answer:**

(b): Write a two argument constructor for *Distance* that accepts two positive integers for feet and inches. Write an access (*get*) method for *feet* and a mutator (*set*) method for *inches*.

**Answer:**

(c): Write a public method *add* that accepts another *Distance* object as a parameter and adds the distance of the other object to the current distance. For example:

```
x1 = new Distance(5,8);    x2 = new Distance (2,7);
x1.add(x2);                // x1 becomes 8 feet, 3 inches
```

**Answer:**

(d): Write a public method *metricDistance* that returns (as a *double*) the number of meters in the distance represented by the object. This method should contain a private variable representing the number of feet in a meter (1 foot = 0.3048 meters).

**Answer:**

(e): Modify the constructor you wrote for part (b) so that if the number of inches is greater than 11, it prints **Illegal number of inches** to the user's screen, and sets *inches* to 11.

**Answer:**

18. Each of the following 3 pieces of code might contain an error. For each part, if there is an error correct it by rewriting just one line of the code in the space provided. If the code has no error, leave the answer space blank.

(a): A two-dimensional array of *int* has been initialized and *total* has been set to 0. This code should calculate *total* as the sum.

```
for (int row = 0; row < numbers.length; row++) {
    for (int col = 0; col < numbers.length; col++)
        total += numbers[row][col]; }
```

**Answer:**

(b):

The following is the *add* method in a *class* called *Fraction*. Read the code and answer the question.

```
public void add (Fraction frac) {
    int a, b, c, d;
    a = this.getNumerator();    b = this.getDenominator();
    c = frac.getNumerator();    d = frac.getDenominator();
    setNumerator(a*d + c*b);    setDenominator(b*d); }
```

Correct the following call to the *add* method.

```
f3 = f1.add(f2);
```

**Answer:**

(c): Given the class definition,

```
public class Pet{
    private String name; private String breed; private int age;
    public setName(String n){ name = n;}
    public setBreed(String b){breed = b;} }
```

Correct the following, if necessary

```
public class PetDemo{
    public static void main(String[] args){
        Pet p1 = null;
        p1.setName("Max");
        p1.setBreed("Dog");
    } }
```

**Answer:**

19. If *str* is a *String* in Java the *str.length()* method gives its length and the *str.charAt(i)* method gives the character of the string at index *i*.

For example,

```
String s1 = "Hello";
int len = s1.length(); //Assigns 5 to len because the length of s1 is 5.
char c = s1.charAt(len-1); //assigns 'o' to c because the character at index 4 = len - 1 is 'o'
```

Use these two string methods to write a complete Java program to find the number of uppercase letters in a sentence typed on the input screen. For example here is a sample run of the program.

```
Enter a sentence: Here Is A Long String.
It has 5 upper-case characters.
```

Use at most 12 lines of code.

**Answer:**

20. What will be printed by this program?

```

public class Question4 {
    static String s1,s2,s3,s4;
    public static void main (String[] args){
        s1 = new String("Kona");
        s2 = new String("Ethiopian");
        s3 = new String("Blue Mountain");
        s4 = new String(s1);

        System.out.println("Match by ==: " + (s1==s4));
        System.out.println("Match by equals() : " + s1.equals(s4));

        coffee(s1,s2);
        System.out.println(mixThem(s2, s3, s4));
    }

    private static void coffee(String a1, String a2) {
        if (a1.compareTo(a2)> 0)
            System.out.println(a2);
        else
            System.out.println(a1);
    }

    private static String mixThem (String b1, String b2, String b3){
        String c1 = b1;
        b1 = b2;
        b1 = new String ("Sulawesi");
        System.out.println(c1);
        System.out.println(b2);
        System.out.println();
        b2 = new String("b3");
        return b2;
    }
}

```

**Answer:**

21. Write static methods for the *class P5* that carry out the indicated tasks in this program.

```

import java.io.PrintWriter;
public class P5 {
    public static void main(String args[]) {
        int a[] [] = {{1, 2, 3, 4}, {0, 1, 2, 3, 6}, {3, 5, 8, 1, 4, 7 }};
        int b[] = {3, 6, 1, 4, 1, 8};
        int x = 1; int y= 2;

        // (a) Return true if at least one of x and y is positive. Here Yes is printed
        if (positive(x, y)) System.out.println( "Yes" );
    }
}

```



```
// (b) Return the sum of the third row. Here 28 is printed.  
    System.out.println(rowSum(a));  
  
// (c) Return the smallest element. Here      1      is printed.  
    System.out.println(smallest(b));  
    }  
}
```

**Answer:**

(a)

(b)

(c)