CSE21

SAMPLE Final
Time: 3 hours
Maximum Points: 300

Name (print):

John Villalvazo

The following precedence table is provided for your use:

Precedence of Operators
( )
- (unary), !, ++, --
*, /, %
+ , - (binary)
<, <=, >, >=
= =, !=
&&
||
=, +=, -=, *=, /=, %=

Otherwise left to right within the expression

Sample Sample Sample Sample Sample Sample Sample

Write all answers in the boxes or on the lines provided.

1. [20 pts] Complete the following Java program that performs a sequential search of the unsorted array x. A sequential search simply starts at the first array element, and compares it with the target value. If it finds the target, the function returns the index where the target was found. The function find_seq searches for a target value. The user enters the target value into y. If the target value is found in the array, the index of the target location in the array is returned to main. If the target value is not found in the array, –1 is returned to main. Complete the statements indicated. Read the provided code carefully.

```
import java.util.*;
public class ProblemOne {

public static void main(String[] args) {
   int y, z;
   int[] x = new int[15];
   Scanner kbd = new Scanner(System.in);

   x[0]=110;x[1]=360;x[2]=45;x[3]=96;x[4]=10;    // set of numbers placed
   x[5]=112;x[6]=14;x[7]=165;x[8]=90;         // in the array
   x[9]=200;x[10]=3;x[11]=42;

   System.out.println( "What number to search for, enter -1 to exit? ");
   y = kbd.nextInt( );;

   while( y != -1)
       {
       z =  find_seq(x, y, x.length);    //x.length because x is an array and it is getting the length of the entire array

       if(z == -1)
          System.out.println( "Sorry, not there.");
       else
          System.out.println( "Found it at index " + z);

      System.out.println( "What number to search for, enter -1 to exit? ");
      y = kbd.nextInt( );;
      }

 } // end of main

 public int find_seq( int[] x, int targ , int numValidVals   //int[]x because its a inter-changeable value and int targ because that is the variable name in the code and int numValidVals is the variable name in the code. )
 {
     int Found = 0, i=0;
     while (Found== 0 && i< numValidVals)
     {
       if (targ != a[i])
            i++; // we want to continue searching for the target number within this loop
       else
            Found = 1; //once found is equal 1 then the return will be whatever index of array a[] is
    }
     if (Found == 0)
       return -1;
     else
       return i ;
   }

}
```

Add a call to the `find_seq` function. This function receives an integer array, a target value, and the number of elements in the array. It returns an integer.

Write the paremeter list for the `find_seq` function

Add code for the true branch of the `if` statement

Add code for the false branch of the `if` statement

Sample Sample Sample Sample Sample Sample Sample

2. [20 pts] Given the program shown below indicate the output in the box below. This is one continuous program. Follow each step carefully. You **will** want to draw the array as partial credit and to help you go through this problem.

```
public class ProblemTwo {

        public static void main(String[] args) {
                int LITTLE = 6, MEDIUM = 10, BIG = 128;

                int i,j, n = 9,temp;
                int[] num= new int[MEDIUM];
                num[0]=9; num[1]=3; num[2]=4; num[3]=8; num[4]=2; num[5]=1;
                num[6]=5; num[7]=6; num[8]=7;


                num[1] = 1;
//i is reinitialized into 5  i = 5;
                while (i < MEDIUM) {
                        num[i] = 2*num[i - 1];
                        i++;
                }

                for (j=0;j<MEDIUM;j++)
                        System.out.print(num[j]+", ");

                }
}
```

num = 4, 2, 10, 12, 14

5 , 6, 7, 8, 9

num[i]= 14

9, 1, 4, 8, 2, 4, 8, 16, 32, 64,

Output

3. [40 pts] Do the following for the ReportCard class specification shown below. The default constructor should set the data member name to "John Doe", year to 12, and idNumber to 12345. Also, allocate the grades array to length 5, and initialize each grade to 'I'. The setGrade function does not return a value, and should receive two parameters: an integer representing which grade to set (0 through 4), and a character representing the letter grade for that class. The function should also perform error checking, to make sure the index number is in the correct range and the grade is valid (A, B, C, D, or F). Finally, the calcGPA method should return a number representing the grade point average, calculated as follows: an A is worth 4 points, B is worth 3, C is worth 2, and D is worth 1 point (assume each grade is for only 1 credit-hour). Return the grade point average as a double.

a) Write the default constructor.
b) Write the setGrade function.
c) Write the calcGPA method.

```
// File: ReportCard.java  Java source code for ReportCard class specification.
import java.util.*;

public class ReportCard {
   private String name;                // student's name
   private int year;                   // student's grade
   private int idNumber;               // student's social security number
   private char[] grades;              // student's grades for this semester
```

```
public ReportCard(){
name = “John Doe”;
year = 12;
idNumber = 12345;
grades = new char[5];
for(int i = 0; i < grades.length; i++)
     grades[i] = “I”;
}


public void  setGrade(int grade, char g){
   if(grade >= 0 && grade <= 4 && (g== 'A' II g== 'B' II g== 'C' II
g== 'D' II g== 'F')){
grades[grade] = g;
}
   }

public double calcGPA(){
double total = 0.0;
for(int i =0; i < grade.length; i++)
       switch (grades[i])
             case ‘A’: total += 4; break;
             case ‘B’: total += 3; break;
             case ‘C’: total += 2; break;
             case ‘D’: total += 1; break;
             case ‘F’: total += 0; break;
}
}

return (total/5.0);
```

```
}
```

Sample Sample Sample Sample Sample Sample Sample

4. [20 pts] Use the information in the ReportCard class on the previous page to write the Java statements described below in the following program segment.

```
import java.util.*;

public class ProblemFour {

public static main(String[] args)
  {
       double finalGPA;
       Scanner kbd = new Scanner(System.in);

       ReportCard johnny = new ReportCard();

       for(int i = 0; i <= 5; i++){
       johnny.setGrade(i, "A");
       }

       finalGPA = johnny.calcGPA();

       System.out.println(fianlGPA);

  }
}
```

Write a Java statement to declare and allocate an instance of the `ReportCard` class that uses the default constructor. Give this instance the name `johnny`.

Write a loop which sets each of the 5 grades in the `johnny ReportCard` to A. You must do this with a loop.

Write Java statements to calculate the overall GPA of the `johnny ReportCard` , store that value in the variable `finalGPA`, and display that value on the screen.

5. [20 pts] Find and fix the bug(s) in the following code. For each bug, indicate the line number, whether it is (C)ompile-time, (R)un-time, or (L)ogical, and explain how to fix the bug.

```
/**
* Return the sum of all the elements of the input array
* @param inp must be a non-null array of type double
* @return the sum of all elements in inp
*/
0 public static double sum(double[] inp) {
1
2    int total;
3
4    for (int e = 0; e <= inp.length; e += 1) {
5       total = inp[e];
6    }
7
8 }
```

line 2 - (L) there should be a double not an int because in the method it requires a return of double

line 4 - (R) the index length can be no more than the length of the array

line 5 - (L) instead of a equals sign there should be a plus equals because it is gathering the total amount of the array

line 7 - (C) there is no return though the method requires a return of type double

line 2 - (L/C) the variable needs to be initialized to zero in order for the variable to be correct

6. [20 pts] The following Java program has several methods. Imagine that the program is run and that you are the computer running the program. Follow mentally each line of code in order of execution. Each time a method is called, write down the name of the method and parameters if any). In the end you will have listed all methods that were executed, and in order of execution. Note that names can be repeated.

```
public class ProblemSix {

  public static void main(String[] args) {
    for (int i = 0; i < 4;i++)
      if (test(i))
        run();

  public static boolean test (int x) {
    if (x == 1 || x == 4)
      return true;
    else if (x == 2)
      return test(x*x);
    else
      return false;
  }

  public static void run() {
  }
}
```

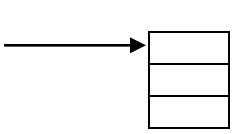
// Your trace here

main()
test(0)
test(1)
run()
test(2)
test(4)
run()
test(3)

Sample Sample Sample Sample Sample Sample Sample

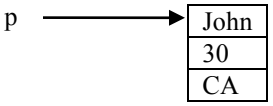
7. [40 pts]You will fill in the follow data and pointers as follows. Imagine we have the Person object from lab with 3 instance variables: name, age and location. If we create it an obiect and pointer at the same time then we would see this:

Person p = new Person();



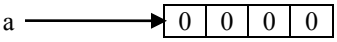
The first entry corresponds to name, second to get and third to location. Then if we set the variable using mutator then the picture would change to the following:

p.setName("John");
p.setAge(30);
p.setLocation("CA");

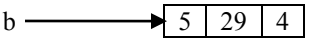


We could also declare arrays like this depending on the number of entries it creates:

int[] a = new int[4];

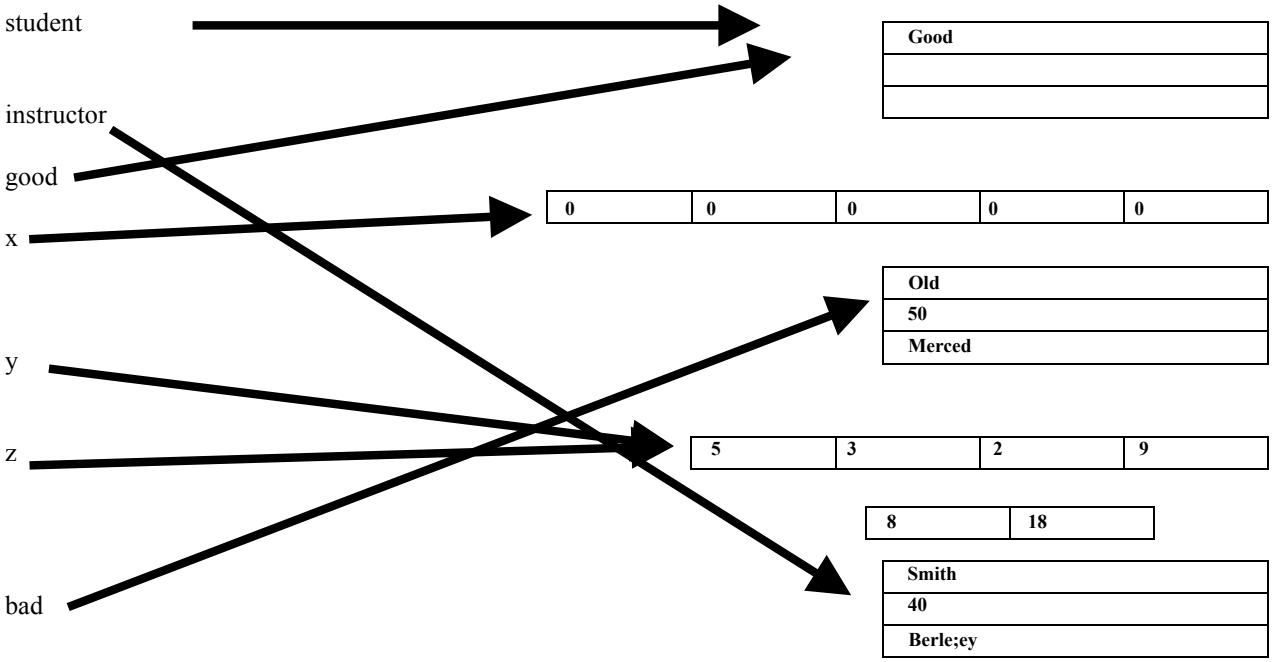


int[] b = {5, 29, 4};



Follow a similar convention for the code below and give the final picture of how the pointers and objects are connected together. You should also fill in the appropriate values into the instances variables or array entries:

```
Person student = new Person();
Person instructor = new Person("Overlord", 50, "Merced");
int[] x = new int[5];
int[] y = new int[2];
int[] z = {5, 3, 2, 9};
Person good = student;
Person bad = instructor;
instructor.setName("Old");
instructor = new Person("Smith", 40, "Berkeley");
good.setName("Good");
y[0] = z[0] + z[1];
y[1] = z[2] * z[3];
z = y;
```



8. [30 pts] Given the following class specification and beginning of a main program that uses the Sigma and Epsilon classes. Pretend Sigma and Epsilon are in separate files. Complete the table below.

```
public class Sigma {
   public void write( ){…}
   public void enterData(int x){…}
   protected int resend(){…}
   private int n;
}

public class Epsilon extends Sigma {
   public void twist() {…}
   public float calcIt(float f){…}
   protected float z;
}
```

```
public class ProblemEight {


public static void main (String[] a) {
   int x=9;
   float y=6.778;
   Epsilon someObject
         = new Epsilon();
      Sigma anotherObject
         = new Sigma();
//more code here…
```

Complete the following table. In the first row list **all method calls** that can legally be made in Java for the anotherObject instance of the Sigma class and for the someObject instance of the Epsilon class. Write the method calls for these objects as they could appear in the main function shown above. Use the variables x and y as arguments if the method calls for any arguments. Do not worry about the order of the calls. You have been given one method call for each instance to start you off. On the second row indicate all the **data members** accessible by the methods of each of the classes, Sigma and Epsilon respectively, and in the third row do the same with the **methods** (functions).

	Sigma class (anotherObject instance)	Epsilon class (someObject instance)
functions that can be called from the client program (main) shown above for the instances of the class.	anotherObject.EnterData(x); **anotherObject.write();**	someObject.CalcIt(y); **someObject.twist();** **someObject.write();** **someObject.enterData(x);**
data members that can be accessed by functions of this class	**n**	**z**
functions that can be called from inside the functions of this class.	**enterData(n)** **write()** **resend()**	**CalcIt(z)** **Twist()** **write()** **enterData(z)** **resend()**

Sample Sample Sample Sample Sample Sample Sample

9. [20 pts] Given the class specification file for the Cuboid class and the derived class Box shown below, indicate whether each of the following statements in the Client.java program are valid. Pretend that Cuboid is in the file Cuboid.java, and Box is in the file Box.java. Circle the word legal or illegal next to the statement.

```
public class Cuboid {
   protected int length;
   protected int width;
   protected int height;
   public void enterData(int l, int w, int h){…}
   public int calcVol() {…}
}
public class Box extends Cuboid {
   private int color;
   public void enterColor(int c){…}
   public void displayColor(){…}
   public int calcSA(){…}
};
```

```
import java.util.*;

public class Client {
       public static void main(String[] args) {
              Cuboid cube = new Cuboid( );
              Box b = new Box( );                    //     Statement 1  legal  illegal
              int len, wid, height, vol_box, vol_cube;
              int color;
              int SA_box;
              Scanner kbd = new Scanner(System.in);

              System.out.println( "Enter length, width, and height: ");
              len=kbd.nextInt(); wid=kbd.nextInt(); height=kbd.nextInt();
              cube.enterData(len, wid, height);
              vol_cube = cube.calcVol();             //     Statement 2  legal  illegal
              System.out.println("Volume of your cuboid is "+ vol_cube);
              System.out.println( "Enter color (1,2,3 = r,b,g; other = black): ");
              color=kbd.nextInt();
              cube.enterColor(color);                //     Statement 3  legal  illegal
              cube.displayColor();                   //     Statement 4  legal  illegal
              System.out.println( "Enter data for your box..." );
              System.out.println( "Enter length, width, and height: ");
              len=kbd.nextInt(); wid=kbd.nextInt(); height=kbd.nextInt();
              b.enterData(len, wid, height);         //     Statement 5  legal  illegal
              System.out.println( "Enter color (1,2,3 = r,b,g; other = black): ");
              color=kbd.nextInt();
              b.enterColor(color);                   //     Statement 6  legal  illegal
              vol_box = b.calcVol();                 //     Statement 7  legal  illegal

              System.out.println("Volume of your box is "+ vol_box);
              System.out.println( "Color of your box is ");
              b.displayColor();                      //     Statement 8  legal  illegal
       }
}
```

Sample Sample Sample Sample Sample Sample Sample

10. [30 pts] The following class generates a random integer between the given minimum and maximum values, inclusively, i.e. [min, max].

```
public class GenInt {
        private int minimum, maximum;

        public GenInt ( int min, int max ) {
                minimum = min;
                maximum = max;
        }

        public int next() {
                return ( minimum
        +(int)(Math.Random()*(maximum-minimum)+0.5) );
        }
}
```

Write a class called GenIntNoRepeat that extends GenInt. This class will behave similarly to GenInt except that it is guaranteed to never generate the same random number twice in a row.

For example, the sequence of integers 2, 8, 3, 3, 4, 1, 2 could be generated by class GenInt however would never be generated by class GenIntNoRepeat because of the two consecutive number 3s.
Be sure to call methods from GenInt in your subclass where appropriate via the super keyword. Consider what values for minimum and maximum will ensure that different numbers can be returned, and whether you should constrain those values in the constructor.

Hints: your derived class will need to re-write both the constructor and the next method. In your new constructor, you will check the validity of the minimum and maximum arguments (just print an error message in case a problem is detected). In your new method, you will need to ensure that it generates a random number in range that is different from the previously generated one, i.e. it will have to continue to try to generate random numbers until a different one is found.

Note: Math.random() returns a double value in [0,1), i.e., greater than or equal to 0.0 and less than 1.0.

```
public class GenIntNoRepeat extends GenInt {
    private int lastVal;
    public GenIntNoRepeat ( int min, int max ) {
        super(min,max);
        lastVal = min;
        if ( max < min ) {
            System.out.println("Error, min/max not valid!");
        }
}
    public int next () {
        int thisVal = super.next();
        while (thisVal == lastVal) {
            thisVal = super.next();
        }
        lastVal=thisVal;
        return thisVal;
    }
}
```

11. [40 pts] Consider the following array definition:

```
int [][] twoD = new int[5][];
twoD[0] = new int[5];
twoD[1] = new int[4];
twoD[2] = new int[3];
twoD[3] = new int[2];
twoD[4] = new int[1];
```

a) Draw the space allocated by the above definition

0,5
1,4
2,3
3,2
4,1

b) Give the output of the following code:

```
System.out.println("Rows of  2D array is " + twoD.length);
for (int i = 0; i < twoD.length; i++)
  System.out.println("Column  " + i + " of 2D array is " + twoD[i].length);
```

Rows of 2D array is 5
Column 0 of 2D array is 5
Column 1 of 2D array is 4
Column 2 of 2D array is 3
Column 3 of 2D array is 2
Column 4 of 2D array is 1

Sample Sample Sample Sample Sample Sample Sample

c) Fill each entry in the twoD array with their corresponding number starting with 0 for the [0][0] entry. You will count from left to right first meaning it will be 1 for [0][1] entry and so on until you reach the end of the first row. Then continue to the next row which would be [1][0]. Do this until all the rows are filled in.

```
int num = 0
for(int i = 0; i < twoD.length; i++)
    for(int j = 0; j < twoD.length; j++)
        twoD[i][j] = num++;
```

d) Ask the user to enter the name of the output file. Open that file for writing. Now output the twoD array's values in the same format as your diagram in a) to the file. That means all the numbers in row 0 has to appear in one line of the output file. Each subsequent row would be on its own line. You may put a space or tab between each number.

```
Scanner input = new Scanner(System.in);
System.out.print("Enter the file name:" );
String filename = input.next();
try {
   FileWriter output = new FileWriter(filename);
   for (int i = 0; i < twoD.length; i++) {
      for (int j = 0; j < twoD[i].length;j++)
         output.write(twoD[i][j] + "\t");
      output.write("\r\n");
}
   output.close();
} catch (Exception e) {
   System.out.println(e);
}
```