Exam 1

CS 0007 Computer Organization

Summer 2020 (2207), MoWe 12:30 – 14:15

(out of 100 points)

Directions: This exam is closed book. You may not use any type of calculator (it is not needed). Put all materials under your desk, including cell/smart phones, smart watches, headphones, calculators, laptops, tablets, etc. All questions are marked with their point value. There should be plenty of workspace provided in the exam booklet, but if you need extra pages, you may use blank pieces of paper.

Show work: Be sure to show all work and turn in any extra pages that you use. If you do not show your work, you may not receive full or partial credit for a correct or wrong answer. Write legibly. If your handwriting cannot be read, then you will not receive credit for an answer.

Pitt ID# _____

Name _____

This page intentionally left blank.

A1 | **Pitt Username:_____ Name:_____ Rec. Time: _____**

Multiple choice

1. Declaring class attributes private can be advantageous, why? Fill-in the correct answer:

   - O It is way easier to implement
   - O Class users can access them directly
   - O Allows the class to validate their values before changing them
   - O All attributes MUST be private

2. If you change the value of a static attribute, what of the following statements is true? Fill-in the correct answer:

   - O It'll only change its value for instances that are already created
   - O The change will only be visible to the instance where it was changed
   - O All instances of the class will see the change
   - O It cannot be changed because it's static

3. Which of the following is the correct way to instantiate an object of type "Class":

   - O Class c = Class();
   - O Class c = new Class();
   - O Class c = new Class;
   - O Class c;

4. In a Java class, static methods can access non-static attributes?

   - O Always
   - O When they are initialized first
   - O Never
   - O If they are public

5. Which of these methods is called automatically when you try to convert the object to a string?

   - O print
   - O toString
   - O object2string
   - O println

6. When an attribute has the same name as an argument, what do we call the effect the latter has on the former?

   - O obscurity
   - O occlusion
   - O shadowing
   - O collision

7. When different valid methods in a class have the same name, we say the method _____ the other. What should be the word in the blank space?

   - O overwhelms
   - O hides
   - O shadows
   - O overloads

8. True or false

	When you create a class you must write a constructor
	It makes no difference if your methods and variables are static or not
	A private method can be called by the user of a class
	If you pass an object as an argument to a method, you are passing a reference. That means that all changes will not be visible outside of the method
	You can write a method in Java that swaps the contents of two variables passed as arguments. HARD QUESTION: WILL NOT BE IN THE FINAL
	In Java, you can have two different methods with the same name if they have different return types
	In Java, you can have two different methods with the same name if they have different arguments
	Objects are complex data types that combine data and the procedures that act on that data.

A1 | **Pitt Username:**_____ **Name:**_____ **Rec. Time:** _____

CODE

9. Suppose the following classes (some details were omitted):

```
public class Slice {
    public boolean isCovered() {…}; // Returns true if the slice is
fully covered
}


public class Sandwich {
    public Slice top;
    public Slice bottom;
}


public class PeanutButter {
    // This class can be used to apply peanut butter to bread slices
    public void openJar() {…}    // Opens the jar of Peanut butter
    public void closeJar() {…}    // Closes the jar of Peanut butter
    public void apply(Slice slice) {…}// Applies a bit of peanut
butter in the slice. Does not guarantee that the slice is covered

}

public class Jelly {
    // This class can be used to apply jelly to bread slices
    public void openJar() {…}  // Opens the jar of Jelly
    public void closeJar() {…}  // Closes the jar of Jelly
    public void apply(Slice slice) {…} // Applies a bit of jelly in
the slice. Does not guarantee that the slice is covered
}
```

A1 | **Pitt Username:_____ Name:_____ Rec. Time: _____**

Complete the following Java code that makes a peanut butter && jelly sandwich. Make sure that:

1. Apply peanut butter to the top slice and jelly to the bottom slice (1+1 points).
2. Both slices are fully covered (2+2 points).
3. The jars are opened before applying, and closed after applying (2+2 points).

```
void main(String[] args) {
    Sandwich sandwich = new Sandwich();
    // Enter your code









    Student student = new Student();
    Luis luis = new Luis();
    luis.eat(sandwich);
    if (luis.isHappy()) {
        student.assignGoodGrade();
    } else {
        student.fail();
    }
}
```

10. Consider the following main function):

```
public class Main {

    public static void main(String[] args) {
        Bike bike = new Bike(“Cannondale”);
        bike.setModel(“Topstone”);
        bike.setColor(Bike.GREEN);
        bike.type = “Gravel”;
        System.out.println(bike.toString());
    }

}
```

Assume the program above produces the following output:

```
Brand: Cannondale
Model: Topstone
Color: Green
Type: Gravel
```

A1 | **Pitt Username:_____ Name:_____ Rec. Time: _____**

Finish implementing class Bike to that the output of running the program matches.

```
public class Bike {
    public static String GREEN = “Green”;
    public static String RED = “Red”;
    public static String YELLOW = “Yellow”;









}
```

11. What is the output of the following program

```
public class Overloaded {

    public Overloaded() {
        this(1);
    }

    public Overloaded(int i) {
        System.out.println(“Hello”);
    }

    public Overloaded(double d) {
        System.out.println(“World”);
        print(1);
    }

    public void print(int i, long l) {
        System.out.println(“print_il”);
        print(3.14);

    }

    public void print(int i) {
        System.out.println(“print_i”);
        print(i, i);
    }

    public void print(double d) {
        System.out.println(“print_d”);
    }

}

class Main{
    public static void main(String[] args) {
        Overloaded o = new Overloaded();
        o.print(1, 2);
        Overloaded o2 = new Overloaded(3.14);
    }
}
```

12. What is the output of the following program?

```
class Main{

  public static int function(int n) {
      if (n < 10) return n;

      String str_n = "" + n;

      int sum = 0;
      for (int i = 0; i < str_n.length(); i++) {
          char letter = str_n.charAt(i);
          int digit = Integer.parseInt(""+letter);
          sum += digit;
      }

      return function(sum);

  }

  public static void main(String[] args) {
      System.out.println( function(1432) );
  }

}
```

13. What is the output of the following program?

```
class MyClass {
    public int x=0;
}

class Main{

  public static void main(String[] args) {

    MyClass m1 = new MyClass();
    MyClass m2 = new MyClass();
    if(m1 == m2) {
       System.out.println("The same!");
    } else {
       System.out.println("Not same!");
    }
    m2 = m1;
    if(m1 == m2) {
       System.out.println("The same!");
    } else {
       System.out.println("Not same!");
    }

    m2.x = 3;
    if(m1 == m2) {
       System.out.println("The same!");
    } else {
       System.out.println("Not same!");
    }

  }

}
```

14. What is the output of the following program?

```
class MyClass {
    private int x=0;
    public MyClass(int x) {
        x = x;
    }
    public int getX() {
        return this.x;
    }
}

class Main{

  public static void main(String[] args) {

    MyClass m = new MyClass(3);
    System.out.println(m.getX());

  }

}
```

15. What is the output of the following program?

```
class Main {

  static int[] array1D = { 1,  2,  3,  4,  5,  6,  7,
                           8,  9, 10, 11, 12, 13, 14,
                          15, 16, 17, 18, 19, 20, 21};

  static int array2D[][] = new int[7][3];

  public static void redistribute() {

      for(int i=0; i<array1D.length; i++) {
          array2D[i/3][i%3] = array1D[i];
      }
  }

  public static void main( String [] args ) {
      redistribute();
      int row = 0;

      while (row < 7) {

        int column = 0;
        while(column < 3) {
          System.out.print(array2D[row][column]+" ");
          column++;
        }

        System.out.println();

        row++;
      }
  }


}
```

This page intentionally left blank.