Summer 2009 CS61b Midterm 1

You have 50 minutes to finish this test. Your exam should contain 6 problems (numbered 0-5) on 8 total pages. This is an open-book test. You may consult any books, notes, or other paper-based inanimate objects available to you. Read the problems carefully. If you find it hard to understand a problem, please ask a question. Partial credit will be awarded where we can, so do try to answer each question. Please write your answers in the spaces provided in the test; if you need to use the back of a page make sure to clearly tell us so on the front of the page.

Good Luck!

0		out of 1 point
1		out of 6 points
2		out of 6 points
3		out of 9 points
4		out of 4 points
5 (part 1)		out of 4 points
5 (part 2)		out of 4 points
Total		out of 34 points

Flip to the last page and write your name!

Birthday: Month: ________ Day: ________

Question #1

a) What is the output of the following code? Write your answer in the box.

```
int[] myArray = {1, 2, 3, 4, 5};
System.out.println(myArray[4]);
myArray = new int[4];
System.out.println(myArray[3]);
```

b) Fill in the blanks below to indicate what is printed by running the main method of Mystery.java shown below. There are no compile-time or run-time errors in this program.

```
public class Mystery {
  public static void mystery1(boolean [] bArray){
      boolean b;
      for (int i = 0; i< bArray.length; i++){
          b = bArray[i];
          b = !b;
      }
      Mystery.mystery2(bArray);   // 1
      bArray[2] = false;
      Mystery.mystery2(bArray);   // 2
      bArray = new boolean[4];
      Mystery.mystery2(bArray);   // 3
  }
  public static void mystery2(boolean [] bArray){
      int i = bArray.length - 1;
      while (i > 0){
          System.out.print(bArray[i] + " ");
          i--;
      }
      System.out.println();
  }
  public static void main(String [] args){
      boolean [] bArray = {true, true, true, true, false};
      Mystery.mystery1(bArray);
      Mystery.mystery2(bArray); // 4
  }
}
```

	Write what is printed
// 1	
// 2	
// 3	
// 4	

____

6

Birthday: Month: ________ Day: ________

Question #2

```
public class ExceptionalStuff {
    public static void crazy(int i) ________________________________{
        if (i == 0){
            System.out.println("1");
            throw new NullPointerException();
            System.out.println("2");
        }
        try {
            System.out.println("3");
            throw new ExceptionA();
            System.out.println("4");
        } catch (Exception e) {
            System.out.println("5");
            throw new ExceptionB();
            System.out.println("6");
        } finally {
            System.out.println("7");
        }
    }
```

a) `ExceptionA`, and `ExceptionB` all extend `Exception`. For the code above to compile, what **must** be added to the blank above? (Circle 0 or more of the words below)

`throws`   `ExceptionA,`   `ExceptionB,`   `NullPointerException`

b) What is printed by `ExceptionalStuff.crazy(0);`? (You do not need to print anything for exceptions.)

c) What is printed by `ExceptionalStuff.crazy(1);` ? (You do not need to print anything for exceptions.)

____
6

Birthday: Month: ________ Day: ________

Question #3

For each example of code, respond whether or not it will compile. If it compiles, please respond whether or not it will run without errors. If it runs without errors and has a return value, please write the return value.

```
interface X {
     int method();
}
class Y implements X {
     int method(){
          return 0;
     }
     private double method(String arg){
          return 12.20;
     }
}
class Z extends Y{
     double method(String arg){
          return 3.14;
     }
}
public class testXYZ{
     public static void main(String[] args) {
```

// Code – Each group of lines is independent	Compiles?		Runs without errors?		Return value?
`(new Y()).method("hi");`	YES	NO	YES	NO	
`(new Z()).method();`	YES	NO	YES	NO	
`((Z) (new Y())).method("yo");`	YES	NO	YES	NO	

`X x1 = new Z();`
`Y y1 = (Z) x1;`
	YES	NO	YES	NO	

`X[] xarr = {new Y(), new X()};`	YES	NO	YES	NO	
`Y[] yarr = {new Y(), new Z()};`	YES	NO	YES	NO	
`((Y) (new Z())).method("hey");`	YES	NO	YES	NO	

`X x2 = new Z();`
`Z z2 = (Y) x2;`
`z2.method();`
	YES	NO	YES	NO	

`X x3 = new Z();`
`x3.method("hello");`
	YES	NO	YES	NO	

```
     }
}
```

____
9

Birthday: Month: ________ Day: ________

Question #4

Fill in the blanks below with legal Java to produce the output indicated in each comment. If it is impossible write “IMPOSSIBLE” in the blank. **You may not create any additional objects!**

```java
public class Parent {
	public void feed(Parent p){
		System.out.println("Parent feed Parent");
	}
	public void feed(Child c){
		System.out.println("Parent feed Child");
	}
}
public class Child extends Parent {
	public void feed(Parent p){
		System.out.println("Child feed Parent");
	}
	public void feed(Child c){
		System.out.println("Child feed Child");
	}
	public static void main(String[] args)
	{
		Parent p = new Child();
```

	// Child feed Child
	// Child feed Parent
	// Parent feed Child
	// Parent feed Parent

```java
		p = new Parent();
```

	// Child feed Child
	// Child feed Parent
	// Parent feed Child
	// Parent feed Parent

```java
	}
}
```

____

4

Birthday: Month: ________ Day: ________

Question #5 (continued on next page)

Below is a modification of code from the `Account` class. Read the syntactically valid code provided and debug the method `removePoorParents()`. This method should remove any parent from the chain of parents that has a balance less than 1,000. An `Account` that has their parent `Account` removed should still be able to access the parent of their former parent `Account` (Assuming that parent `Account` has a balance of 1000 or greater.)

a) Fill in the `main` method below with code to demonstrate the logical error in `removePoorParents()`. Also fill in the blanks to explain the error.

```
public class Account {
  private Account myParent;
  private int myBalance;
  public Account(int balance, Account parent) {
        this.myBalance = balance;
        this.myParent = parent;
  }
  public void removePoorParents() {
        if (this.myParent != null) {
              if (this.myParent.myBalance < 1000) {
                    this.myParent = this.myParent.myParent;
                    if(this.myParent == null){
                          return;
                    }
              }
              this.myParent.removePoorParents();
        }
  }
  public static void main(String[] args) {




/* At this point __________________ is _____________________________

 * but it should be ________________________________________________

 */
  }
}
```

____

4

Birthday: Month: ________ Day: ________

Question #5 (continued from previous page)

b) Modify the `removePoorParents()` method below to fix the bug you demonstrated in part a).

```
public void removePoorParents() {

    if (this.myParent != null) {

        if (this.myParent.myBalance < 1000) {

            this.myParent = this.myParent.myParent;

            if(this.myParent == null){

                return;

            }

        }

        this.myParent.removePoorParents();

    }

}
```

___

4

Last Name:	
First Name:	
Login:	cs61bl-
TA (circle):	Jon 8-11 Kaushik 11-2 David 2-5 George 6-9
Birthday:	Month: _____ Day: _____

Please fill out the names of people around you

	YOU are Here	

Question #0
Write the month and day of your birthday on every page! (worth 1 pt)