CS 1331 Fall 2016 Exam 3 Part 1

Fall 2016

ANSWER KEY

- Failure to properly fill in the information on this page will result in a deduction of up to 5 points from your exam score.
- Signing signifies you are aware of and in accordance with the **Academic Honor Code of Georgia Tech** and that you will not discuss this exam with other students.
- Calculators and cell phones are NOT allowed.
- This is an object-oriented programming test. Java is the required language. Java is case-sensitive. DO NOT WRITE IN ALL CAPS. A Java program in all caps will not compile. Good variable names and style are required. Comments are not required.

Question	Points per Page	Points Lost	Points Earned	Graded By
Page 1	14	-	=	
Page 2	2	-	=	
Page 3	6	-	=	
Page 4	8	-	=	
Page 5	8	-	=	
Page 6	10	-	=	
Page 7	0	-	=	
Page 8	0	-	=	
TOTAL	48	-	=	

1. **Multiple Choice** Circle the letter of the correct choice.

[2] (a) Which of the following exceptions is/are throwable exceptions?

A. FileNotFoundException
B. NullPointerException
C. IndexOutOfBoundsException
D. None of the above are throwable exceptions
E. All of the above are throwable exceptions

[2] (b) Which of the following exceptions is/are checked exceptions?

A. FileNotFoundException
B. NullPointerException
C. IndexOutOfBoundsException
D. None of the above are checked exceptions
E. All of the above are checked exceptions

[2] (c) What is the *highest* superclass of all exception classes?

A. Throwable
B. Exception
C. RuntimeException
D. Error
E. Object

2. **Short Answer**

[4] (a) The method header for the `addAll` method in a `Set` is `public boolean addAll(Collection<? extends E> c)` What kind of objects can I put in a `List` that I want to pass to this method?

Solution: Any object that is of type `E` or a subclass of `E`.

[4] (b) Given the class header `public class HashMap<K, V>` create an instance of this class that could be used to map names to ages (assume name is a `String` and age is an `int`).

Solution: `HashMap<String, Integer> namesToAges = new HashMap<String, Integer>();`

 Points available: 14 - points lost: _____ = points earned: _____. Graded by: _____

3. Given the following code:

```
import java.util.ArrayList;
import java.util.Comparator;
import java.util.Collections;

public class Cat implements Comparable {
    public int age;
    public String name;

    public Cat(int age, String name) {
      this.age = age;
      this.name = name;
    }

    private static class CatComparator<T extends Cat> implements Comparator<T> {
      public int compare(T a, T b) {
        if (a.compareTo(b) == 0) {
          return a.name.compareTo(b.name);
        }
        return a.compareTo(b);
      }
    }

    public int compareTo(Object other) {
        Cat that = (Cat) (other);
        return this.age - that.age;
    }

    public static void main(String[] args) {
      ArrayList<Cat> cats = new ArrayList<>();
      cats.add(new Cat(8, "steph"));
      cats.add(new Cat(4, "adi"));
      cats.add(new Cat(5, "chris"));
      cats.add(new Cat(2, "taylor"));
      cats.add(new Cat(4, "sam"));

      CatComparator<Cat> catComp = new CatComparator<>();
      Collections.sort(cats, catComp);

      for (Cat cat: cats) {
        System.out.print(cat.name + " ");
      }
      System.out.println("\n");
    }
}
```

[2] (a) What will the main method above print out?

A. taylor sam adi chris steph
B. adi chris sam steph taylor
C. taylor adi sam chris steph
D. Nothing will print. This code will not compile.

 Points available: 2 - points lost: _____ = points earned: _____. Graded by: _____

[2] (b) If Cat "andy" has an age of 6, and Cat "john" has an age of 9, how will they be naturally ordered?

A. andy, john

B. john, andy

C. the ordering cannot be determined

[4] 4. What will be printed out by the code below?

```
public class Cheer {
    static void go() throws Throwable {
        throw new Throwable("Go Jackets!");
    }
    static void fight() throws Throwable {
        go();
        System.out.println("Fight!");
    }
    public static void main(String[] args) {
        try {
            fight();
        } catch (Throwable t) {
            System.out.println(t.getMessage());
        }
        System.out.println("Still cheering.");
     }
}
```

```
Go Jackets!
Still Cheering.
```

 Points available: 6 - points lost: _____ = points earned: _____. Graded by: _____

5. Consider the code below for a Car class

[4] (a) Write the necessary code to make `Car` objects have a natural ordering by year.

```
public class Car _______________________________{
    private String make;
    private int year;

    public Car(String make, int year) {
        this.make = make;
        this.year = year;
    }
    public String getMake() {
        return make;
    }
    public int getYear() {
        return year;
    }

    public boolean equals(Object other) { return super.equals(other); }

    public int hashCode() { return super.hashCode(); }

    //write code below









}
```

[4] (b) Write a single statement (not necessarily in one line of code) to sort a `List` of `Car` objects, called cars, alphabetically by make.

Solution: `Collections.sort(cars, (a, b) -> a.getMake().compareTo(b.getMake()));`

 Points available: 8 - points lost: _____ = points earned: _____. Graded by: _____

6. Circle the `Collection` type that is best suited for each scenario. You may use a type multiple times or no times at all. One collection per question.

[2] (a) You want to manage prices for a grocery store of fruits. Each fruit has an associated price. Every fruit of the same type has the same price.

A. List
B. Set
C. Map

[2] (b) You want to schedule students. A student cannot be enrolled for 2 instances of the same class. Your scheduler needs to know what classes a Student is enrolled in.

A. List
B. Set
C. Map

[2] (c) You want to model a backpack's content. The backpack can hold multiple items and we want to distinguish items of the same type.

A. List
B. Set
C. Map

[2] (d) You want to keep track of your friends' first names and their birthdays. Some of the names are repeated, but have different birthdays. You want to be able to sort your friends by their birthdays and also by their names.

A. List
B. Set
C. Map

 Points available: 8 - points lost: _____ = points earned: _____. Graded by: _____

[10] 7. **Complete the code by filling in the blanks so that it will compile.**

```
public class Book implements  Comparable <Book > {
    private String title;
    public Book(String title) { this.title = title; }
    public String toString() { return title; }
    public int compareTo(Book book ) {
        return title.compareTo(book.title);
    }
}
```

```
public class BookShelf implements Iterable<Book> {
    private List<Book> books;
    public BookShelf(Collection<Book> books) { books = new ArrayList<>(books);}
    public void printAll() {for (Book b: books) {System.out.println(b);}
    public Iterator<Book> iterator() {
        Collections.sort(books, new BookCritic());
        return ...;
        // assume that a valid iterator over the books List is returned
        // that iterates over the books in the original order
    }
}
```

 Points available: 10 - points lost: _____ = points earned: _____. Graded by: _____