CS 2334: Programming Structures and Abstractions
Exam 2
November 4, 2015

General instructions:

- Please wait to open this exam booklet until you are told to do so.
- This examination booklet has 14 pages. You also have been issued a bubble sheet.
- Write your name, university ID number and date, and sign your name below. Also, write your name and ID number on your bubble sheet, and fill in the bubbles for your ID.
- You may have up to five pages of your own notes. No electronic devices or books may be used.
- The exam is worth a total of 137 points. Your grade counts for 10% of your final grade.
- You have 1.25 hours to complete the exam. Be a smart test taker: if you get stuck on one problem go on to the next.
- Use your bubble sheet to answer all multiple-choice questions. Make sure that the question number and the bubble row number match when you are answering each question.

On my honor, I affirm that I have neither given nor received inappropriate aid in the completion of this exam.

Signature: _____

Name: _____

ID Number: _____

Date: _____

Question	Points	Score
Generics	25	
Lists, Queues and Stacks	40	
Sets and Maps	30	
Graphical User Interfaces	30	
Enumerated Data Types	12	
Total:	137	

A C A B B C A C
C B A A C B A B
B C A B C C A B

Part I. Generics

1. (5 points) Will the following code compile?

```
public class Foo<E>  {
    int a;

    public Foo(int a){
        this.a = a;
    }
}
```

A. Yes B. No C. Answer not shown

2. (5 points) Generic types are checked:

A. never **B. compile time** C. run time D. compile time and run time
E. Answer not shown

3. (5 points) Will the following code compile?

```
public class Foo<E> implements Comparable<E> {
    E a;

    public Foo(E a){
        this.a = a;
    }

    public int compareTo(E b){
        return a.compareTo(b);
    }
}
```

A. Yes **B. No** C. Answer not shown

4. (5 points) Consider the following code:

```
LinkedList<Integer> LL1 = new LinkedList<Integer>();
LinkedList<Double> LL2 = new LinkedList<Double>();

display(LL1);
display(LL2);
```

What is the proper method header for a method that will display the list that is passed to it? (this method header must work in both cases)

A.
```
public static void display(LinkedList<? super Number> list){...};
```

B.
```
public static void display(LinkedList<? implements Number> list){...};
```

C.
```
public static void display(LinkedList<? extends Number> list)↩
    {...};
```

D.
```
public static void display(LinkedList<Number> list){...};
```

E. None of the above will compile

5. (5 points) Will the following code compile?

```
public class Foo<E extends Comparable<E>> {
    E a;

    public Foo(E a){
        this.a = a;
    }

    public int compareTo(E b){
        return a.compareTo(b);
    }
}
```

A. Yes B. No C. Answer not shown

Part II. Lists, Queues and Stacks

6. (6 points) What is printed by this block of code?

```
PriorityQueue<String> q = new PriorityQueue<String>();


q.add("sews");
q.add("clothes");
q.add("sue");
q.add("crow's");

while(!q.isEmpty()){
  System.out.print(q.remove() + " ");
}
```

A. sews clothes sue crow's
B. clothes crow's sews sue
C. crow's sue clothes sews
D. sue sews crow's clothes
E. Answer not shown

7. (6 points) What is printed by this block of code?

```
Stack<String> s = new Stack<String>();

s.push("battle");
s.push("paddle");
s.push("puddle");
s.push("beetle");
s.push("tweetle");

while(!s.isEmpty()){
  System.out.print(s.pop() + " ");
}
```

A. tweetle puddle paddle beetle battle
B. battle paddle puddle beetle tweetle
C. battle beetle paddle puddle tweetle
D. tweetle beetle puddle paddle battle
E. Answer not shown

Consider the following class definition:

```
public class Item implements Comparable<Item>{
    private String description;
    private double cost;

    public static class ItemComparator implements Comparator<Item>{
        public int compare(Item i1, Item i2) {
            if(i1.cost < i2.cost) return -1;
            if(i1.cost > i2.cost) return 1;
            return 0;
        }
    }

    public static class ItemComparator2 implements Comparator<Item>{
        private int sign;

        public ItemComparator2(int sign){
            this.sign = sign;
        }

        public int compare(Item i1, Item i2) {
            if(i1.cost < i2.cost) return -sign;
            if(i1.cost > i2.cost) return sign;
            return 0;
        }
    }

    public Item(String description, double cost) {
        this.description = description;
        this.cost = cost;
    }

    public int compareTo(Item i) {
        int val = description.compareTo(i.description);
        if(val == 0){
            if(cost < i.cost) return -1;
            if(cost > i.cost) return 1;
        }
        return val;
    }

    public String toString(){
        return description + "(" + cost + ")";
    }

    public static void displayList(List list){
        for(Object o: list){
            System.out.print(o + ", ");
        }
        System.out.println("");
    }
```

And consider the following main method (part of the same class):

```
    public static void main(String[] args){
        LinkedList<Item> list = new LinkedList<Item>();

        list.addFirst(new Item("mouse", 22));
        list.addLast(new Item("keyboard", 35));
        list.addFirst(new Item("disk", 49));
        list.addLast(new Item("mouse", 16));

        Item.displayList(list); // Line 1

        Collections.sort(list);
        Item.displayList(list); // Line 2

        Collections.sort(list, new ItemComparator());
        Item.displayList(list); // Line 3

        Collections.sort(list, new ItemComparator2(-1));
        Item.displayList(list); // Line 4
    }
}
```

8. (7 points) What is printed on Line 1 of the output?

A. mouse(22), keyboard(35), disk(49), mouse(16),
B. disk(49), keyboard(35), mouse(22), mouse(16),
C. mouse(16), mouse(22), keyboard(35), disk(49),
D. disk(49), mouse(22), keyboard(35), mouse(16),
E. disk(49), keyboard(35), mouse(16), mouse(22),

9. (7 points) What is printed on Line 2 of the output?

A. mouse(22), keyboard(35), disk(49), mouse(16),
B. disk(49), keyboard(35), mouse(22), mouse(16),
C. mouse(16), mouse(22), keyboard(35), disk(49),
D. disk(49), mouse(22), keyboard(35), mouse(16),
E. disk(49), keyboard(35), mouse(16), mouse(22),

10. (7 points) What is printed on Line 3 of the output?

A. mouse(22), keyboard(35), disk(49), mouse(16),
B. disk(49), keyboard(35), mouse(22), mouse(16),
C. mouse(16), mouse(22), keyboard(35), disk(49),
D. disk(49), mouse(22), keyboard(35), mouse(16),
E. disk(49), keyboard(35), mouse(16), mouse(22),

11. (7 points) What is printed on Line 4 of the output?

A. mouse(22), keyboard(35), disk(49), mouse(16),
B. disk(49), keyboard(35), mouse(22), mouse(16),
C. mouse(16), mouse(22), keyboard(35), disk(49),
D. disk(49), mouse(22), keyboard(35), mouse(16),
E. disk(49), keyboard(35), mouse(16), mouse(22),

Part III. Sets and Maps

12. (6 points) What is printed by this block of code?

```
TreeMap<Integer,Integer> map = new TreeMap<Integer,Integer>();

map.put(13, 19);
map.put(19, 29);
map.put(3, 7);
map.put(29, 37);
map.put(7, 13);

for(int i: map.keySet()){
  System.out.print(map.get(i) + " ");
}
```

A. 7 13 19 29 37 B. 13 19 3 29 7 C. 3 7 13 19 29 D. 19 29 7 37 13
E. Answer not shown

13. (6 points) What is printed by this block of code?

```
TreeMap<Integer,Integer> map = new TreeMap<Integer,Integer>();

map.put(13, 19);
map.put(3, 7);
map.put(19, 29);
map.put(7, 13);
map.put(29, 37);

System.out.println(map.get(map.get(7)));
```

A. 3 B. 7 C. 13 **D. 19** E. Answer not shown

14. (6 points) What is printed by this block of code?

```
HashSet<Integer> set = new HashSet<Integer>();
set.add(7);
set.add(16);
set.add(42);
set.add(7);
set.add(37);

System.out.println(set.size() + "-" + set.contains(3));
```

A. 5-true B. 4-true C. 5-false **D. 4-false** E. Answer not shown

Consider the following code block:

```
TreeMap<Integer, ArrayList<String>> m =
                    new TreeMap<Integer, ArrayList<String>>();
ArrayList<String> l1 = new ArrayList<String>();
ArrayList<String> l2 = new ArrayList<String>();

l1.add("Foo");
l1.add("Bar");

m.put(2332, l1);
m.put(1860, l2);

l2.add("baz");
l2.add("end");
```

15. (6 points) What is printed by this following block of code?

```
System.out.println(m.get(1860).get(1));
```

A. Foo B. Bar C. baz **D. end**
E. Answer not shown or Null Pointer Exception

16. (6 points) What is printed by this following block of code?

```
for(Integer i: m.keySet()){
  System.out.print(m.get(i).get(0) + " ");
}
```

A. baz Foo B. Foo baz C. Bar end D. end Bar
E. Answer not shown or Null Pointer Exception

Part IV. Graphical User Interfaces

Consider the following program:

```
public class GUI1 extends JFrame{
    private JPanel panel;
    private JButton b1;
    private JButton b2;
    private JTextField t1;
    private int i = 0;

    public GUI1(){
        super("Save Window");

        panel = new JPanel();
        add(panel);
        panel.setLayout(new GridLayout(2,0));

        b1 = new JButton("press here");
        b2 = new JButton("press there");
        t1 = new JTextField("start");

        panel.add(b1);
        panel.add(b2);
        panel.add(t1);

        b1.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent arg0) {
                i++;
                t1.setText("" + i);
            }
        });

        b2.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent arg0) {
                i = 0;
            }
        });

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        pack();
        setVisible(true);
    }

    public static void main(String[] args){
        GUI1 f = new GUI1();
    }
}
```

17. (7 points) When the program first starts, what is displayed?

A.

B.

C.

D.

E. Answer not shown

18. (6 points) If the buttons are pressed in the following order, what is displayed in the JTextField?

here-here-here-there

A. start B. 1 C. 2 **D. 3** E. Answer not shown

19. (6 points) If the buttons are pressed in the following order, what is displayed in the JTextField?

here-here

A. start B. 0 **C. 2** D. 3 E. Answer not shown

20. (6 points) If the buttons are pressed in the following order, what is displayed in the JTextField?

here-here-here-there-here

A. start **B. 1** C. 2 D. 3 E. Answer not shown

21. (5 points) Which of the following statements is true?

A. GUI1 *is-a* JFrame and *has-a* JPanel

B. GUI1 *has-a* JFrame and *has-a* JPanel

C. GUI1 *has-a* JFrame and *is-a* JPanel

D. JFrame *has-a* GUI1 and GUI1 *is-a* JPanel

E. Answer not shown or multiple statements are true

Part V. Enumerated Data Types

Consider the following class definition:

```
public enum CSCourse {
  CS1323("Intro", 4), CS2334("Abstractions", 4), CS2413("Structures", 3);

  private int units;
  private String name;

  private CSCourse(String name, int units){
    this.name = name;
    this.units = units;
  }

  public String toString(){
    return name + "(" + units + ");";
  }

  public int getUnits(){
    return units;
  }
}
```

22. (6 points) What is printed by the following line of code?

```
System.out.println(CSCourse.CS2413);
```

A. Structures B. CS2413 C. CSCourse.CS2413 **D. Structures(3);**
E. This line would not compile

23. (6 points) What is the correct implementation of a method that tests whether a course is CS2334?

A.
```
public static boolean testCourse(CSCourse course){
    return course == 2334;
}
```

B.
```
public static boolean testCourse(CSCourse ↩
    course){
      return course == CSCourse.CS2334;
}
```

C.
```
public static boolean testCourse(CSCourse course){
    return course.equals("Abstractions");
}
```

D.
```
public static boolean testCourse(CSCourse course){
    return course == "CS2334";
}
```

E. None of the implementations are correct