CS 2334: Programming Structures and Abstractions
Exam 2
November 6, 2017

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.
- Fill in the identifying information below (signature, name, ID and date) Also, write your name and ID number on your bubble sheet, and fill in the bubbles for your ID. **Exams without this information will be assigned a grade of zero.**
- You may have up to two 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.
- Other than **this** page, you may tear any other page out of this booklet that does not contain numbered answers.
- If you cannot effectively erase erroneous answers from the bubble sheet, please clearly cross them out.

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	28	
Lists, Queues and Stacks	32	
Sets and Maps	42	
Graphical User Interfaces	25	
Enumerated Data Types	10	
Total:	137	

Part I. Generics

1. (4 points) Consider the following code blocks:

```
public static <T1, T2 extends T1> void add1(T2 obj, List<T1> list)
{
   list.add(obj);
}

public static <T1, T2 extends T1> void add2(T1 obj, List<T2> list)
{
   list.add(obj);
}
```

Which of the following is true?

A. add1() and add2() would both compile
B. Only add1() would compile
C. Only add2() would compile
D. Neither method would compile

2. (4 points) Will the following code compile?

```
public class TwoThings<T>
{
    private T thing1;
    private T thing2;

    public TwoThings(T thing1, T thing2)
    {
        this.thing1 = thing1;
        this.thing2 = thing2;
    }

    public String toString()
    {
        return thing1 + "+" + thing2;
    }
}
```

A. Yes B. No

3. (4 points) What is output from the following code block:

```
ArrayList<Integer> list = new ArrayList<Integer>();

System.out.println(list instanceof List);
```

Which of the following is true?

A. true B. false C. This code does not compile

4. (4 points) Will the following code compile?

```
public class MyClass1<E extends Number>
{
    private E val;

    public MyClass1(E val)
    {
        this.val = val;
    }

    public E doubleValue()
    {
        return val * 2;
    }
}
```

A. Yes **B. No**

5. (4 points) Will the following code compile?

```
public class OneThing<E extends Comparator<E>> implements Comparable<E>
{
  public static int compare(E v1, E v2)
  {
    return v1.compareTo(v2);
  }
}
```

A. Yes **B. No**

6. (4 points) Will the following code compile?

```
public class MyNumber<T> extends Number
{
    private T num;

    public MyNumber(T num)
    {
        this.num = num;
    }
}
```

A. Yes **B. No**

7. (4 points) Will the following code compile?

```
public class LinkedElement<E>
{
    private E data;
    private LinkedElement<E> next;

    public LinkedElement(LinkedElement<E> next)
    {
        this.data = new E();
        this.next = next;
    }

    public E getData()
    {
        return data;
    }
}
```

A. Yes **B. No**

Part II. Lists, Queues and Stacks

Consider the following class definition:

```
public class Item implements Comparable<Item>
{
    private String s;
    private Integer v;

    public Item(String s, int v)
    {
        this.s = s;
        this.v = v;
    }

    public int compareTo(Item i)
    {
        return i.s.compareTo(s);
    }

    public static class ItemComparator1 implements Comparator<Item>
    {
        public int compare(Item a, Item b)
        {
            return a.v.compareTo(b.v);
        }
    }

    public static class ItemComparator2 implements Comparator<Item>
    {
      public int compare(Item a, Item b)
      {
        int ret = b.v.compareTo(a.v);
        if(ret == 0)
        {
          return a.s.compareTo(b.s);
        }
        else
        {
          return ret;
        }
      }
    }

    public String toString()
    {
        return this.s + this.v;
    }

    public static void display(List<Item> list)
    {
        for(Item i: list)
        {
            System.out.print(i + ",");
        }
        System.out.println("");
      }
}
```

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

```
1     public static void main(String[] args)
2     {
3         LinkedList<Item> list = new LinkedList<Item>();
4
5         list.add(new Item("a", 8));
6         list.add(new Item("d", 6));
7         list.add(new Item("z", 1));
8         list.add(new Item("m", 7));
9
10        display(list);
11
12        Collections.sort(list);
13        display(list);
14
15        Collections.sort(list, new ItemComparator1());
16        display(list);
17
18        Collections.sort(list, new ItemComparator2());
19        display(list);
20    }
```

8. (5 points) What is printed by Line 10?

A. a8, d6, m7, z1,
B. a8, d6, z1, m7,
C. a8, m7, d6, z1,
D. z1, d6, m7, a8,
E. z1, m7, d6, a8,

9. (5 points) What is printed by Line 13?

A. a8, d6, m7, z1,
B. a8, d6, z1, m7,
C. a8, m7, d6, z1,
D. z1, d6, m7, a8,
E. z1, m7, d6, a8,

10. (5 points) What is printed by Line 16?

A. a8, d6, m7, z1,
B. a8, d6, z1, m7,
C. a8, m7, d6, z1,
D. z1, d6, m7, a8,
E. z1, m7, d6, a8,

11. (5 points) What is printed by Line 19?

A. a8, d6, m7, z1,
B. a8, d6, z1, m7,
C. a8, m7, d6, z1,
D. z1, d6, m7, a8,
E. z1, m7, d6, a8,

12. (4 points) What is printed by this block of code?

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

s.push (10);
s.push (5);
s.push (42);
s.pop ();
s.pop ();
s.push (3);
s.pop ();
s.push (2);
s.push (7);
s.pop ();

int sum;
for(sum = 0; !s.isEmpty(); sum = s.pop())
{
};
System.out.println(sum);
```

A. 6
B. 10
C. 12
D. 17
E. Answer not shown

13. (4 points) What is printed by this block of code?

```
Stack<Integer> s = new Stack<Integer> ();
s.push (10);
s.push (3);
s.peek ();
s.push (7);
s.push (4);
s.peek ();

System.out.println (s.pop ());
```

A. 3
B. 4
C. 7
D. 10
E. Answer not shown

14. (4 points) What is printed by this block of code?

```
LinkedList<String> list = new LinkedList<String> ();

list.addFirst("Bob");
list.removeLast();
list.addLast("Ronald");
list.addLast("Skip");
list.addFirst("Jose");
list.addLast("Bob");
list.removeFirst();

for(String s: list)
{
  System.out.print(s + " ");
}
```

A. Bob Ronald Skip Bob
B. Jose Ronald Skip
C. Jose Ronald Skip Bob
D. Ronald Skip Bob
E. Answer not shown

Part III. Sets and Maps

Consider the following code block:

```
1   HashMap<String,Integer> map1 = new HashMap<String, Integer>();
2   TreeMap<Integer, String> map2 = new TreeMap<Integer, String>();
3
4   map1.put("Pan", 17);
5   map1.put("Dione", 7);
6   map1.put("Atlas", 78);
7   map1.put("Callisto", 42);
8
9   map2.put(42, "Dione");
10  map2.put(7, "Callisto");
11  map2.put(78, "Dione");
12  map2.put(17, "Dione");
13  map2.put(42, "Atlas");
14
15  System.out.println(map1.get("Atlas"));
16
17  System.out.println(map1.get(map2.get(7)));
18
19  System.out.println(map2.get(map1.get(map2.get(map1.get("Callisto")))));
20
21  Iterator<Integer> it = map2.keySet().iterator();
22  it.next();
23  System.out.println(it.next());
```

15. (4 points) What is printed by line 15?

A. 7 B. 17 C. 42 **D. 78** E. Answer not shown

16. (4 points) What is printed by line 17?

A. 7 B. 17 **C. 42** D. 78 E. Answer not shown

17. (4 points) What is printed by line 19?

A. Atlas B. Callisto **C. Dione** D. Pan E. Answer not shown

18. (4 points) What is printed by line 23?

A. 7 **B. 17** C. 42 D. 78 E. Answer not shown

Consider the following block of code:

```
1  Set<String> set1 = new TreeSet<String>();
2  Set<String> set2 = new TreeSet<String>();
3
4  set1.add("Adama");
5  set1.add("Baltar");
6
7  set2.add("Starbuck");
8  set2.add("Adama");
9
10 set1.retainAll(set2);
11 System.out.println(set1);
12
13 set2.addAll(set1);
14 System.out.println(set2);
```

19. (5 points) What is printed by line 11?

A. [Adama]
B. [Adama, Baltar]
C. [Adama, Starbuck]
D. [Baltar]
E. [Starbuck]

20. (5 points) What is printed by line 14?

A. [Adama]
B. [Adama, Baltar]
C. [Adama, Starbuck]
D. [Baltar]
E. [Starbuck]

Consider the following code block:

```
1   TreeSet<String> set = new TreeSet<String>();
2
3   set.add("Alice");
4   set.add("Hatter");
5   set.add("Mouse");
6   set.add("White Rabbit");
7   set.add("Mouse");
8
9   System.out.println(set.contains("mouse"));
10  System.out.println(set.contains("White Rabbit"));
11  System.out.println(set.contains("Caterpillar"));
12
13  set.remove("Hatter");
14  System.out.println(set.size());
```

21. (4 points) What is printed by line 9?
A. true **B. false** C. Going to be late D. Answer not shown

22. (4 points) What is printed by line 10?
A. true B. false C. Answer not shown

23. (4 points) What is printed by line 11?
A. true **B. false** C. Answer not shown

24. (4 points) What is printed by line 14?
A. 2 **B. 3** C. 4 D. 5 E. Answer not shown

Part IV. Graphical User Interfaces

Consider the following program:

```
public class GUI extends JFrame
{
  private JButton button1;
  private JButton button2;
  private JLabel label1;
  private JLabel label2;
  private JLabel label3;

  public GUI()
  {
    this.setLayout(new GridLayout(0, 2));

    button1 = new JButton("One Fish");
    button2 = new JButton("Two Fish");
    label1 = new JLabel("L 1");
    label2 = new JLabel("L 2");
    label3 = new JLabel("L 3");

    this.add(button1);
    this.add(button2);
    this.add(label1);
    this.add(label2);
    this.add(label3);

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

    button1.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e)
      {
        label1.setText("Red");
        label2.setText("");
        label3.setText("Fish");
        button1.setBackground(new Color(0, 255, 0));
      }
    });

    button2.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e)
      {
        label2.setText("Blue");
        label1.setText("");
        label3.setText("FISH");
        button1.setBackground(new Color(255, 0, 0));
      }
    });
  }

  public static void main(String[] args)
  {
    GUI gui = new GUI();

  }
}
```

25. (4 points) True or False? label2 is a **Container**.

A. True B. False

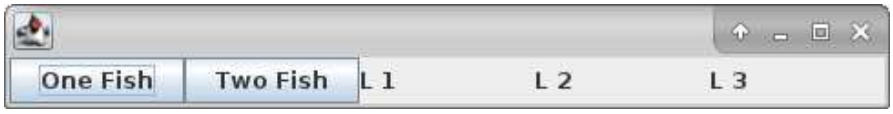
26. (4 points) If the **One Fish** button is pressed, what color is the **One Fish** button?
A. Red **B. Green** C. Blue D. Yellow E. Gray

27. (4 points) If the **Two Fish** button is pressed, what is displayed (in label order)?
A. “”-Red-Fish B. Red-Fish-“” C. Blue-“”-FISH
D. “”-Blue-FISH E. Blue-Fish-“”

28. (4 points) True or False? An **Event** object describes how the program should respond to an event.
A. True **B. False**

29. (4 points) True or False? **JButton** is a concrete class.
A. True B. False

30. (5 points) When the program first starts, what is displayed?

A.


B.


C.


D.


E. Answer not shown

Part V. Enumerated Data Types

Consider the following enumerated data type definition:

```
public enum CSCourse
{
    CS1323("Java I", 4), CS2334("Java II", 4), CS2413("Data Structures", 3);

    private int units;
    private String title;

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

    public String toString(){
        return this.name() + " (hours = " + units * 3 + ")";
    }

    public static void main(String[] arg)
    {
        System.out.println(CSCourse.CS2334);
    }
}
```

31. (5 points) What is the correct implementation of a method that tests whether a course is the introductory course (CS 1323)?

A.
```
public static boolean isIntro(CSCourse course){
    return course == 1323;
}
```

B.
```
public static boolean isIntro(CSCourse course){
    return course.equals("Java I");
}
```

C.
```
public static boolean isIntro(CSCourse course){
    return course == "CS1323";
}
```

D.
```
public static boolean isIntro(CSCourse course){
    return course == CSCourse.CS1323;
}
```

E. None of the implementations are correct

32. (5 points) What is printed by the following line of code?

```
System.out.println(CSCourse.CS2334);
```

A. Java II (hours = 12) B. Java II C. CS2334

D. CS2334 (hours = 12) E. CSCourse.CS2334