CS 2334: Programming Structures and Abstractions
Exam 2
November 2, 2016

General instructions:

- Please wait to open this exam booklet until you are told to do so.
- This examination booklet has 15 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.
- 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.
- 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	29	
Lists, Queues and Stacks	32	
Sets and Maps	35	
Graphical User Interfaces	25	
Enumerated Data Types	16	
Total:	137	

Part I. Generics

1. (4 points) Will the following code compile?

```
public class MyString<E>
{
    private String strg;

    public MyString(E val)
    {
        strg = val.toString();
    }
}
```

A. Yes B. No

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

```
public class NegatedNumberArray<E extends Number> extends ArrayList<Number>
{
    public NegatedNumberArray()
    {
        super();
    }

    public boolean add(E val)
    {
        return super.add(- val.doubleValue());
    }
}
```

A. Yes B. No

3. (4 points) Will the following code compile?

```
public class Coordinate<E super Number>
{
    E x;
    E y;

    public Coordinate(E x, E y)
    {
        this.x = x;
        this.y = y;
    }

    public double l1Norm()
    {
        return x.doubleValue() + y.doubleValue();
    }
}
```

A. Yes **B. No**

4. (5 points) Consider the following program:

```
1  public class StackTest
2  {
3      public static <E> void popPush(Stack<? extends E> s1, Stack<? super E> s2)
4      {
5          s2.push(s1.pop());
6      }
7
8      public static void main(String[] args)
9      {
10         Stack<Integer> s1 = new Stack<Integer> ();
11         Stack<Number> s2 = new Stack<Number> ();
12
13         popPush (s1, s2);
14         popPush (s2, s1);
15     }
16 }
```

Which of the following is true?

A. There is an error at line 5
B. There is an error at line 13
C. There is an error at line 14
D. There are no errors

5. (4 points) Generic types are checked:

A. compile time and run time **B. compile time** C. run time D. never

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

```
public class MyNumber2<E extends Comparable<E>> implements Comparator<E>
{
    private E value;

    public MyNumber2 (E x)
    {
        value = x;
    }

    public int compare(MyNumber2 n1, MyNumber2 n2)
    {
        return n1.value.compareTo(n2.value);
    }
}
```

A. Yes B. No

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

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

    public MyNumber(E x)
    {
        val = x.doubleValue();
    }
}
```

A. Yes B. No

Part II. Lists, Queues and Stacks

8. (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.push (3);
s.pop ();
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

9. (4 points) What is printed by this block of code?

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

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

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

Consider the following class definition:

```
public class Thing implements Comparable<Thing>
{
    private Double score;
    private Double subScore;

    public Thing(Double score, Double subScore)
    {
        this.score = score;
        this.subScore = subScore;
    }

    public int compareTo(Thing t)
    {
        return score.compareTo(t.score);
    }

    public static class ThingComparator1 implements Comparator<Thing>
    {
        public int compare(Thing thing1, Thing thing2)
        {
            Double prod1 = thing1.score * thing1.subScore;
            Double prod2 = thing2.score * thing2.subScore;

            return prod1.compareTo(prod2);
        }
    }

    public static class ThingComparator2 implements Comparator<Thing>
    {
        private int flip;

        public ThingComparator2(int flip) {
            this.flip = flip;
        }

        public int compare(Thing thing1, Thing thing2)
        {
            return flip * thing1.subScore.compareTo(thing2.subScore);
        }
    }

    public String toString()
    {
        return score.toString() + "=" + subScore.toString();
    }

    public static void displayAll(List<Thing> list)
    {

        for(Thing t: list)
        {
            System.out.print(t + ", ");
        }
        System.out.println();
    }
}
```

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

```
1     public static void main (String[] args)
2     {
3         ArrayList<Thing> list1 = new ArrayList<Thing> ();
4         list1.add(new Thing(4.0, 7.0));
5         list1.add(new Thing(1.0, 42.0));
6         list1.add(new Thing(9.0, 200.0));
7         list1.add(new Thing(1.0, 8.0));
8
9         displayAll(list1);
10
11        Collections.sort(list1);
12
13        displayAll(list1);
14
15        Collections.sort(list1, new ThingComparator1());
16
17        displayAll(list1);
18
19        Collections.sort(list1, new ThingComparator2(-5));
20
21        displayAll(list1);
22    }
```

10. (5 points) What is printed by Line 9?

A. 9.0=200.0, 1.0=42.0, 1.0=8.0, 4.0=7.0,
B. 1.0=42.0, 1.0=8.0, 4.0=7.0, 9.0=200.0,
C. 4.0=7.0, 1.0=42.0, 9.0=200.0, 1.0=8.0,
D. 4.0=7.0, 1.0=8.0, 1.0=42.0, 9.0=200.0,
E. 1.0=8.0, 4.0=7.0, 1.0=42.0, 9.0=200.0,

11. (5 points) What is printed by Line 13?

A. 9.0=200.0, 1.0=42.0, 1.0=8.0, 4.0=7.0,
B. 1.0=42.0, 1.0=8.0, 4.0=7.0, 9.0=200.0,
C. 4.0=7.0, 1.0=42.0, 9.0=200.0, 1.0=8.0,
D. 4.0=7.0, 1.0=8.0, 1.0=42.0, 9.0=200.0,
E. 1.0=8.0, 4.0=7.0, 1.0=42.0, 9.0=200.0,

12. (5 points) What is printed by Line 17?

A. 9.0=200.0, 1.0=42.0, 1.0=8.0, 4.0=7.0,
B. 1.0=42.0, 1.0=8.0, 4.0=7.0, 9.0=200.0,
C. 4.0=7.0, 1.0=42.0, 9.0=200.0, 1.0=8.0,
D. 4.0=7.0, 1.0=8.0, 1.0=42.0, 9.0=200.0,
E. 1.0=8.0, 4.0=7.0, 1.0=42.0, 9.0=200.0,

13. (5 points) What is printed by Line 21?

A. 9.0=200.0, 1.0=42.0, 1.0=8.0, 4.0=7.0,
B. 1.0=42.0, 1.0=8.0, 4.0=7.0, 9.0=200.0,
C. 4.0=7.0, 1.0=42.0, 9.0=200.0, 1.0=8.0,
D. 4.0=7.0, 1.0=8.0, 1.0=42.0, 9.0=200.0,
E. 1.0=8.0, 4.0=7.0, 1.0=42.0, 9.0=200.0,

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

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

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

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

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

Part III. Sets and Maps

15. (5 points) What is printed by this block of code?

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

map.put("baz", 78);
map.put("foo", 32);
map.put("bar", 55);
map.put("foobar", 2332);

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

A. 55 78 2332 32
B. 32 55 78 2332
C. 2332 78 55 32
D. 2332 32 78 55
E. Answer not shown

16. (5 points) What is printed by this block of code?

```
HashMap<String, String> map = new HashMap<String, String>();

map.put("Mungo", "Bungo");
map.put("Milo", "Minto");
map.put("Berylla", "Mungo");
map.put("Ivy", "Berylla");
map.put("Laura", "Bungo");
map.put("Rufus", "Milo");
map.put("Bungo", "Bilbo");

String s = "Ivy";
while (map.containsKey(s))
{
  s = map.get(s);
}

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

A. Berylla B. Bungo C. Ivy **D. Bilbo** E. Answer not shown

17. (5 points) What is printed by this block of code?

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

map.put ("foo", 32);
map.put ("baz", 189);
map.put ("bar", 55);
map.put ("baz", 78);
map.put ("foobar", 2332);

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

A. foo baz bar baz foobar
B. foo foobar bar baz baz
C. foo foobar bar baz
D. foobar foo baz bar
E. foobar foo baz baz bar

18. (5 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(16);
set.remove(37);
set.add(7);
set.add(37);
set.remove(16);

System.out.println(set.size() + "-" + set.contains(7));
```

A. 3-true B. 3-false C. 4-true D. 5-false E. Answer not shown

Consider the following code block:

```
TreeMap<Integer, Integer> map1 = new TreeMap<Integer, Integer>();
TreeMap<Integer, Integer> map2 = new TreeMap<Integer, Integer>();

TreeMap<Integer, TreeMap<Integer, Integer>> mapMap =
        new TreeMap<Integer, TreeMap<Integer, Integer>> ();

mapMap.put(17, map1);
mapMap.put(23, map2);

map1.put(97, 23);
map1.put(44, 15);

map2.put(17, 4);
map2.put(23, 88);
```

19. (5 points) What is printed by this following block of code?

```
System.out.println(mapMap.get(23).get(17));
```

A. 4 B. 15 C. 23 D. 88 E. Answer not shown

20. (5 points) What is printed by this following block of code?

```
System.out.println(mapMap.get(17).get(23));
```

A. 4 B. 15 C. 23 D. 88 **E. Answer not shown**

21. (5 points) What is printed by this following block of code?

```
System.out.println(mapMap.get(map1.get(97)).get(23));
```

A. 4 B. 15 C. 23 **D. 88** 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 label;

  public GUI()
  {
    this.setLayout(new BorderLayout());
    JPanel panel = new JPanel();
    this.add(panel);

    button1 = new JButton("fox");
    button2 = new JButton("box");
    label = new JLabel("???");

    this.add(button1, BorderLayout.WEST);
    this.add(button2, BorderLayout.EAST);
    this.add(label, BorderLayout.SOUTH);

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

    button1.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e)
      {
        label.setText("socks");
      }
    });

    button2.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e)
      {
        label.setText("knox");
      }
    });
  }

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

22. (4 points) If the **box** button is pressed, what is displayed in the gray part of the box?

    A. ??? B. fox C. box **D. knox** E. socks

23. (4 points) If the **fox** button is pressed, what is displayed in the gray part of the box?

    A. ??? B. fox C. box D. knox **E. socks**

24. (5 points) When the program first starts, what is displayed?

A. 

B. 

C. 

D. 

E. Answer not shown

25. (4 points) True or False? Button2 is a **Container**.
A. True B. False

26. (4 points) True or False? **ActionListener** is a concrete class.
A. True **B. False**

27. (4 points) True or False? An **EventListener** describes the properties of an event.
A. True **B. False**

Part V. Enumerated Data Types

Consider the following enumerated data type definition:

```
public enum PersonType
{
  FLESHER(true, true),
  GLEISNER(false, true),
  CITIZEN(false, false);

  private boolean biological;
  private boolean physical;

  private PersonType(boolean biological, boolean physical)
  {
    this.biological = biological;
    this.physical = physical;
  }

  public boolean getBiological()
  {
    return biological;
  }

  public boolean getPhysical()
  {
    return physical;
  }

  public static boolean isFlesher(PersonType p)
  {
    return p == PersonType.FLESHER;
  }

  public static boolean isGleisner(PersonType p)
  {
    return p.name().equals("GLEISNER");
  }

  public static boolean isCitizen(PersonType p)
  {
    return p.equals("CITIZEN");
  }

  public String toString()
  {
    return this.name() + "(" + physical + ")";
  }
}
```

28. (4 points) What is printed by the following lines of code?

```
PersonType p = PersonType.FLESHER;
System.out.println(isFlesher(p));
```

A. true B. false

29. (4 points) What is printed by the following lines of code?

```
PersonType p = PersonType.CITIZEN;
System.out.println(isCitizen(p));
```

A. true **B. false**

30. (4 points) What is printed by the following lines of code?

```
PersonType gleisner = PersonType.GLEISNER;
System.out.println(gleisner);
```

A. gleisner(false) B. gleisner(true) C. GLEISNER(false)
D. GLEISNER(true) E. Answer not shown

31. (4 points) What is printed by the following lines of code?

```
PersonType p = PersonType.GLEISNER;
System.out.println(isGleisner(p));
```

A. true B. false