CS 2334: Programming Structures and Abstractions
Final Exam
December 14, 2017

General instructions:

- Please wait to open this exam booklet until you are told to do so.
- This examination booklet has 26 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 274 points. Your grade counts for 20% of your final grade.
- You have two 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
Objects and Inheritance	28	
Abstract Classes and Interfaces	20	
IO, Exceptions and Error Handling	34	
Generics	25	
Java Collections Framework	20	
Enumerated Data Types	17	
Graphical User Interfaces and Graphics	26	
Recursion	44	
Ethics	60	
Total:	274	

Part I. Objects and Inheritance

Consider the following class definitions:

```
public class A
{
    protected int val;
    protected String s;

    public A(int val, String s) {
        this.val = val;
        this.s = s;
    }

    public String toString() {
        return s + getVal();
    }

    public int getVal() {
        return val;
    }
}

public class B extends A
{
    protected int val;

    public B(int val, String s) {
        super(val + 1, s);
        this.val = val;
    }

    public B(int val) {
        super(val × 2, "foo");
    }

    public int getVal() {
        return val × super.val;
    }
}

public class C extends B
{
    protected String s;

    public C(int val, String s) {
        super(val + 1, s + s);
        this.s = s;
    }

    public int getVal() {
        return val;
    }
}
```

1. (6 points) What is printed by this block of code?

```
A a = new A(3, "Skip");
System.out.println(a);
```

A. Skip0 B. 0Skip **C. Skip3** D. 3Skip E. Answer not shown

2. (6 points) What is printed by this block of code?

```
B b = new B(5, "Fred");
System.out.println(b);
```

A. Fred5 B. Fred6 C. Fred25 **D. Fred30**
E. Answer not shown

3. (6 points) What is printed by this block of code?

```
B b = new B(11);
System.out.println(b);
```

A. foo0 B. foo11 C. foo22 D. foo242
E. Answer not shown

4. (6 points) What is printed by this block of code?

```
C c = new C(1, "Bob");
System.out.println(c);
```

A. Bob1 B. BobBob1 C. Bob2 **D. BobBob2** E. Answer not shown

5. (4 points) What is printed by this block of code?

```
String s1 = "thx";
String s2 = "tHx";
if(s1.toLowerCase().equals(s2))
{
  System.out.println("Yes:" + s2);
}
else
{
  System.out.println("No:" + s1);
}
```

A. No:thx B. No:tHx C. Yes:thx D. Yes:tHx
E. Compilation error or answer not shown

Part II. Abstract Classes and Interfaces

6. (4 points) Any abstract class could be implemented instead as an interface.
   A. True **B. False**

7. (4 points) Will the following abstract class compile?

```
public abstract class Foo
{
    abstract int x;
    abstract int add(int input);
    abstract int getX();
}
```

   A. Yes **B. No**

8. (4 points) An abstract class being extended by another class is best described as what type of relationship?
   A. Contains-a B. Has-a **C. Is-a** D. Uses-a E. Answer not shown

9. (4 points) Any abstract class can implement any interface.
   A. True B. False

10. (4 points) Which of the following statements is true?
    A. KeyHandler extends KeyAdapter B. KeyHandler implements KeyAdapter
    C. KeyAdapter extends KeyHandler **D. KeyAdapter implements KeyHandler** E. None are true

Part III. IO, Exceptions and Error Handling

Consider the following program:

```
public class ExceptionTest
{
  public static int compute2(int x)
  {
    if(x > 10)
    {
      try
      {
        return compute1(x - 20);
      }
      catch(ArithmeticException e)
      {
        return 0;
      }
    }
    else
    {
      return compute1(x - 5);
    }
  }

  public static int compute1(int x)
  {
    if(x < 3)
    {
      return 5 / (x + 3);
    }
    else
    {
      throw new IllegalStateException("Bad");
    }
  }

  public static int main(String[] args)
  {
    ....
  }
}
```

11. (6 points) What is printed by this code block?

```
try
{
  System.out.println(compute1(6));
}
catch(Exception e)
{
  System.out.println(e);
}
```

A. 0 B. 0.55555... C. 1 **D. IllegalStateException**
E. ArithmeticException

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

```
try
{
  System.out.println(compute1(-3));
}
catch(Exception e)
{
  System.out.println(e);
}
```

A. 0 B. 0.83333... C. 1 D. IllegalStateException
E. ArithmeticException

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

```
try
{
  System.out.println(compute1(-1));
}
catch(Exception e)
{
  System.out.println(e);
}
```

A. 1 B. 1.25 **C. 2** D. IllegalStateException E. ArithmeticException

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

```
try
{
  System.out.println(compute2(25));
}
catch(Exception e)
{
  System.out.println(e);
}
```

A. 0 B. 1 C. 2 **D. IllegalStateException** E. ArithmeticException

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

```
try
{
  System.out.println(compute2(22));
}
catch(Exception e)
{
  System.out.println(e);
}
```

A. 0 **B. 1** C. 2 D. IllegalStateException E. ArithmeticException

16. (4 points) Which interface class is most appropriate for storing a sequence of primitive values to a file?

A. BufferedInputStream B. DataInputStream **C. DataOutputStream**
D. ObjectInputStream E. ObjectOutputStream

Part IV. Generics

17. (4 points) What is the most appropriate prototype for a static mergeSort() method that can sort a list of any sortable object type?

A.
```
public static <T> ArrayList<T> mergeSort(ArrayList<T> list)
```

B.
```
public static ArrayList<Comparable> mergeSort(ArrayList<Comparable> list)
```

C.
```
public static ArrayList mergeSort(ArrayList list)
```

D.
```
public static <T extends Comparable<T>> ArrayList<T>
                                mergeSort(ArrayList<T> list)
```

E. None of the implementations are appropriate

18. (4 points) Suppose that we would like to implement a method that will take as input a Set containing any Mammal. What would the method prototype be?

A.
```
public static void processSet(Set<?> set)
```

B.
```
public static void processSet(Set<Mammal> set)
```

C.
```
public static void processSet(Set<? implements Mammal> set)
```

D.
```
public static void processSet(Set<? extends Mammal> set)
```

E. None of the implementations are correct

19. (4 points) Given the following type declaration:

```
Map<Integer, Map<ArrayList<String>, Integer>> m2;
```

What is the return type of:

```
m2.get(-1).get("baz")
```

A. Integer B. String C. ArrayList D. Map **E. Answer not shown**

20. (4 points) Given the following type declaration:

```
Map<String, ArrayList<Map<Integer, String>>> m;
```

What is the return type of:

```
m.get("foo").get(5)
```

A. Integer B. String C. ArrayList **D. Map** E. Answer not shown

21. (4 points) Given the following type declaration:

```
Map<Integer, Map<ArrayList<String>, Integer>> m2;
```

What is the return type of:

```
m2.get(7).values().iterator().next()
```

A. Integer B. String C. ArrayList D. Map E. Answer not shown

22. (5 points) Will the following code compile?

```
public class Foo<E extends Comparable<E>> implements Comparable<Foo<E>>,
                                                     Comparator<E>
{
    private E x;

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

    public  int compareTo(Foo<E> f) {
        return x.compareTo(f.x);
    }

    public int compare(E e1, E e2)
    {
        return e1.compareTo(e2);
    }
}
```

A. Yes B. No

Part V. Java Collections Framework

23. (6 points) What is printed by this block of code?

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

map.put(1, 7);
map.put(2, 3);
map.put(4, 1);
map.put(5, 7);
map.put(7, 3);
map.put(12, 2);
map.put(14, -4);

int val = 4;

while(map.containsKey(val))
{
  val += map.get(val);
}

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

A. 1 B. 5 C. 7 **D. 10** E. Answer not shown

24. (6 points) Consider the following class definition:

```
public class MyComparator implements Comparator<Integer>
{
    public int compare(Integer i1, Integer i2)
    {
        return -i1.compareTo(i2);
    }

}
```

What is printed by this block of code?

```
TreeSet<Integer> set = new TreeSet<Integer>(new MyComparator());
set.add(5);
set.add(2);
set.add(7);
set.add(3);
set.add(2);

for(int i: set)
{
  System.out.print(i + " ");
}

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

A. 2 3 5 7
B. 7 5 3 2
C. 2 2 3 5 7
D. 5 2 7 3 2
E. Answer not shown

25. (4 points) Consider an **ArrayList<String>** object instance that currently contains n items and has a capacity m, such that $m > n$. How many reference copies are performed by *list.add*("Nosedive")?

A. 0 **B. 1** C. n D. n+1 E. Answer not shown

26. (4 points) Consider an **ArrayList<String>** object instance that currently contains n items and has a capacity m, such that $m > n$. How many reference copies are performed by *list.add*(0,"Playtest")?

A. 0 B. 1 C. n **D. n+1** E. Answer not shown

Solution: I have decided to accept C as well since "copy" is a ambiguous as to whether the new entry might involve a copy (given the way that we have talked about things in class).

Part VI. Enumerated Data Types

Consider the following class definition:

```
public enum Value
{
  P(52), Q(41), R(32), S(22), T(0);

  private double cutoff;

  private Value(double cutoff)
  {
    this.cutoff = cutoff;
  }

  public boolean isGreaterThan(Value v)
  {
    return this.cutoff > v.cutoff;
  }

  public static Value findValue(double x)
  {
    Value v = T;

    for(Value v2: Value.values())
    {
      if(x >= v2.cutoff && v.cutoff < v2.cutoff)
      {
        v = v2;
      }
    }

    return v;
  }
}
```

and these variable initializations:

```
Value v1 = Value.R;
Value v2 = Value.S;
```

27. (4 points) What is printed by this block of code?

```
System.out.println(v1 == v2);
```

A. true **B. false**

28. (4 points) What is printed by this block of code?

```
System.out.println(v1.isGreaterThan(v2));
```

A. true B. false

29. (5 points) What is printed by this block of code?

```
System.out.println(findValue(23));
```

A. P B. Q C. R **D. S** E. T

30. (4 points) True or False? The different values of an enumerated data type make up a proper set.

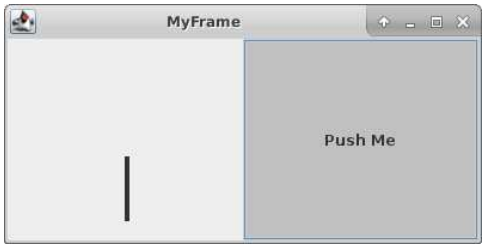
A. True B. False

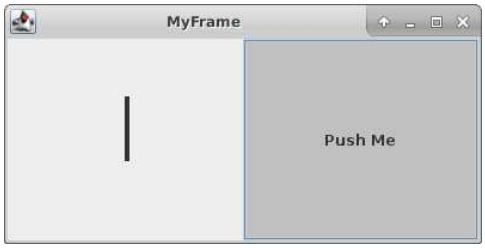
Part VII. Graphical User Interfaces and Graphics

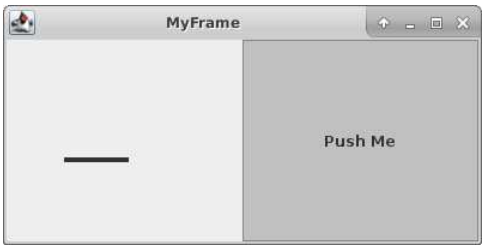
Consider the following program:

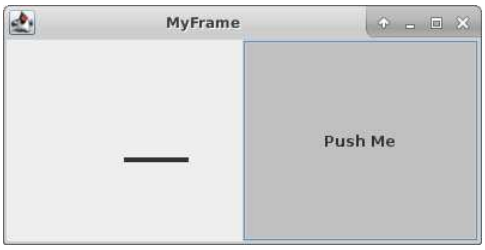
```
public class MyFrame extends JFrame {
  private int width = 4;
  private double orient = 0.0;
  private final int x = 100;
  private final int y = 100;

  public class MyPanel extends JPanel
  {
    protected void paintComponent(Graphics g)
    {
      Graphics2D g2 = (Graphics2D) g;
      super.paintComponent(g);
      BasicStroke stroke = new BasicStroke(MyFrame.this.width);
      g2.setStroke(stroke);
      g2.drawLine(x, y, (int) (x + 50 × Math.cos(MyFrame.this.orient)),
                        (int) (y + 50 × Math.sin(MyFrame.this.orient)));
    }
  }[mathescape=true]
  public MyFrame()
  {
    super("MyFrame");
    this.setLayout(new GridLayout(0,2));
    MyPanel panel = new MyPanel();
    this.add(panel);

    JButton button = new JButton("Push Me");
    button.setBackground(Color.lightGray);
    button.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        orient += Math.PI / 2.0;
        repaint();
        MyFrame.this.requestFocus();
      }
    });
    this.add(button);

    this.addKeyListener(new KeyAdapter() {
      public void keyTyped(KeyEvent e){
        char c = e.getKeyChar();
        switch (c) {
          case 'a':
            width = 1;
            break;
          case 'b':
            width = 9;
            break;
        }
        repaint();
      }
    });
    this.setFocusable(true);
    this.requestFocus();
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(400,200);  // Set size of the entire frame, including menu bar
    setVisible(true);
  }
  public static void main(String[] args)
  {
    MyFrame frame = new MyFrame();
  }
}
```

31. (6 points) When the program first starts, what is displayed?

A. 

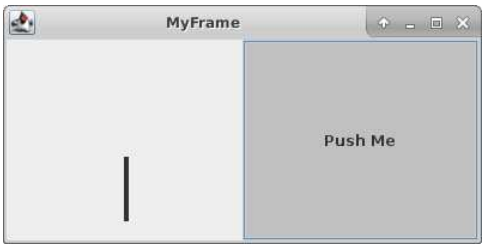
B. 

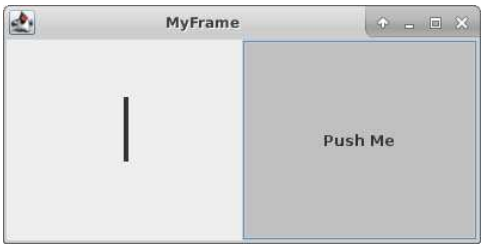
C. 

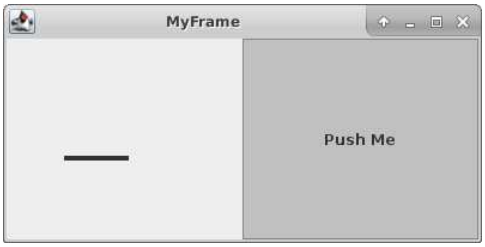
D. 

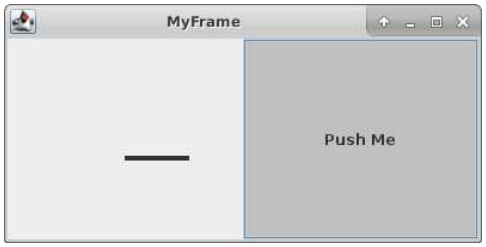
E. Answer not shown

32. (6 points) After the program starts and the user presses the *Push Me* button once, what is displayed?

A. 

B. 

C. 

D. 

E. Answer not shown

33. (6 points) After the user presses the "b" key, what happens to the thickness of the line?

A. Decreases B. Stays the same **C. Increases**

34. (4 points) A JList is which of the following (mark all that apply):

A. A Component B. A List C. An Event **D. An Event generator** **E. An Event handler**

35. (4 points) A ListModel is which of the following (mark all that apply):

A. A Component **B. A List** C. An Event **D. An Event generator** E. An Event handler

Part VIII. Recursion

Consider the following class definition:

```
public class Node extends LinkedList<Node>
{
  private boolean visited;

  private static LinkedList<Node> nodes = new LinkedList<Node>();

  public Node()
  {
    nodes.add(this);
  }

  public static void clearAll()
  {
    for(Node n: nodes)
    {
      n.visited = false;
    }
  }

  public int process()
  {
    if(this.visited)
    {
      return 0;
    }
    else
    {
      this.visited = true;
      int i = 1;
      for(Node n: this)
      {
        i += n.process();
      }
      return i;
    }
  }
}
```

36. (6 points) What is printed by this block of code?

```
public static void main(String[] args)
{
  Node n0 = new Node();

  Node.clearAll();
  System.out.println(n0.process());
}
```

A. 0 **B. 1** C. 2 D. 3 E. Answer not shown

37. (6 points) What is printed by this block of code?

```
public static void main(String[] args)
{
  Node n1 = new Node();
  Node n2 = new Node();

  n2.add(n1);

  Node.clearAll();
  System.out.println(n2.process());
}
```

A. 0 B. 1 **C. 2** D. 3 E. Answer not shown

38. (6 points) What is printed by this block of code?

```
public static void main(String[] args)
{
  Node n1 = new Node();
  Node n2 = new Node();
  Node n3 = new Node();

  n1.add(n2);
  n2.add(n1);
  n3.add(n2);

  Node.clearAll();
  System.out.println(n1.process());
}
```

A. 1 **B. 2** C. 3 D. 4 E. Answer not shown

39. (6 points) What is printed by this block of code?

```
public static void main(String[] args)
{
  Node n1 = new Node();
  Node n2 = new Node();
  Node n3 = new Node();
  Node n4 = new Node();
  Node n5 = new Node();

  n1.add(n2);
  n2.add(n1);
  n2.add(n4);
  n3.add(n2);
  n4.add(n5);
  n5.add(n3);

  Node.clearAll();
  System.out.println(n1.process());
}
```

A. 1 B. 2 C. 3 D. 4 **E. Answer not shown**

Consider the following recursive method:

```
1   public static ArrayList<String> findStrings(int n)
2   {
3     ArrayList<String> list = new ArrayList<String>();
4
5     if(n > 0)
6     {
7       ArrayList<String> list2 = findStrings(n - 1);
8
9       for(String s: list2)
10      {
11        list.add(s + "0");
12        if(s.charAt(s.length() - 1) == '0')
13        {
14          list.add(s + "1");
15        }
16      }
17    }
18    else
19    {
20      list.add("0");
21      list.add("1");
22    }
23
24    return list;
25  }
```

40. (5 points) Which set of lines implement the base case?
A. 7 – 16 B. 14 **C. 20 – 21** D. 24 E. Answer not shown

41. (5 points) What is the length of the returned array in the base case?
A. 0 B. 1 **C. 2** D. 4 E. Answer not shown

42. (5 points) What is printed by the following code block?

```
System.out.println(findStrings(2).get(0));
```

A. 00 B. 10 **C. 000** D. 100 E. Answer not shown

43. (5 points) What is printed by the following code block?

```
System.out.println(findStrings(3).get(3));
```

A. 0010 **B. 0100** C. 0110 D. 0101 E. Answer not shown

Part IX. Ethics

44. (4 points) What does the Health Insurance Portability and Accountability Act of 1996 (HIPAA) say about a patient's ability to control the sharing of specific medical information with other entities?

A. The patient has no control
B. The patient must opt-in to sharing
C. The patient must opt-out of sharing

45. (4 points) True or False? The use of cookies can decrease privacy.

A. True B. False

46. (4 points) True or False? The *Family Educational Rights and Privacy Act* requires that any of your college records be made available to your family.

A. True **B. False**

Solution:

- Once you turn 18, all of the rights afforded by FERPA transfer to you.

47. (4 points) True or False? The *Creative Commons* license can be configured to allow removal of the original author's name.

A. True B. False

48. (4 points) True or False? A *Creative Commons* license that allows derivative works and commercial use can be changed in a derivative work to disallow commercial use.

A. True B. False

XimpleWare offers two versions of a parser for the eXtensible Markup Language (XML). The first version is freely available under the GNU Public License (GPL); the second version has a commercial license that involves payment for its use. A company called *Versata* incorporated (by static linking) the GPL'd version of the code into its own product. Versata then licensed its program commercially to a company called *Ameriprise*.

49. (5 points) Are these actions allowed under the GNU Public License? Explain your answer.

Solution:

- As described, the answer is: Yes, these actions are allowed. The GPL allows a receiver of code to incorporate this code into its own products.
- There are restrictions here. This is technically a derivative work, so Versata is obligated to release its code.

50. (5 points) A subcontractor of Ameriprise *decompiled* the code that it had received from Versata. Decompilation is the process of taking compiled code and transforming it to source code (in Java, this would mean going from .class files to .java files). Is this allowed under the GPL? Why or why not?

Solution:

- Yes. According to the GPL, Ameriprise and its subcontractor have access to the source code (though Versata should have released it).

51. (5 points) Is Ameriprise allowed to modify the code that it decompiled? Why or why not?

Solution:

- Yes. The GPL allows derivative works and is transitive.

52. (5 points) At some point in the process, the text describing the license was removed from the code. Does this present a problem? Why or why not?

Solution:

- Yes, this presents a problem. The GPL license requires all copies of the code, including derivative works to carry with it the GPL license.

Two weeks ago, the US Supreme Court heard a case in which an individual had been convicted of a string of robberies of Radio Shack stores. This individual had been identified as a suspect by multiple witnesses. His involvement was confirmed, in part, because the FBI obtained 127 days of cell phone location data showing that he was in the vicinity of each of the robberies around the time that they occurred.

In order to provide cell phone service, this location data is computed every time that a cell phone connects to the network (e.g., when making a call or downloading data). Because these location data are collected and stored by a third party (the cell phone service providers), the FBI argues that *third party doctrine* is applicable. This legal precedent requires that a subpoena be issued for information requested from a third party; this is a lower threshold than a *warrant* in that no judicial officials are required to review the request.

53. (5 points) Which component(s) of the US guiding documents could the Supreme Court use to justify disallowing access to data in this way (i.e., by changing the requirement from a subpeona to a warrant)?

Solution:

- The fourth amendment prevents unreasonable search and seizure of property owned by a person. This has been broadly interpreted as providing for a digital right to privacy.

54. (5 points) In part, the *third party doctrine* stems from a 1979 Supreme Court case (Smith v. Maryland) in which the contents of a "pen register" were provided to law enforcement. The pen register is a record of which phone numbers are dialed by which other phone numbers. The court stated that an individual does not have a reasonable expectation of privacy for information that s/he voluntarily turns over to a third party. List two arguments that can be made against applying this ruling

to the cell phone location case.

Solution:

- In this particular case, it could be argued that the individual was not aware that location data were being collected.
- Also in this case, the individual did not explicitly "hand over" location information to the 3rd party. Instead, this information was synthesized by the 3rd party.
- We can also argue that the world has fundamentally changed since 1979: a lot more information is being collected about us by 3rd parties, and law enforcement and other entities have a much greater ability to synthesize information from many different sources. This creates an imbalance between privacy and the capabilities of law enforcement that was not intended.
- We explicitly share information with many 3rd parties and this sharing is protected by law. For example, the HIPAA law explicitly calls on health care providers to only share information when approved by us.

55. (5 points) What is the general term for information such as a dialed phone number?

Solution:

- Meta-data

56. (5 points) Should the Supreme Court leave the current procedures in place or should they impose further restrictions on obtaining this type of data (by requiring a warrant)? Why or why not?

Solution: The answers may go either way.

- Leave in place: this is still an important tool for law enforcement. Adding restrictions would reduce the information that they can receive and increase the time required to obtain the data.
- More restrictions: the ability to collect and sift through data has increased dramatically since the third party doctrine was established. We need to bring back some balance.