

Name: _____ Andrew Id: _____

15-121 Fall 2024 Assessment 2

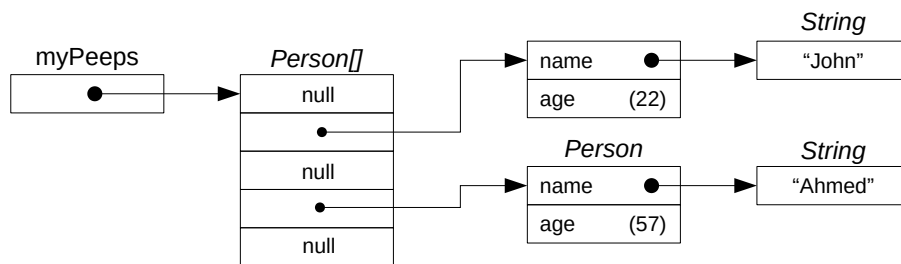
Up to 50 minutes. No calculators, no notes, no books, and no computers. Show your work!

1. Short Answer

Answer the following in 1 sentence; longer answers will be marked incorrect.

- (a) (2 points) When declaring an instance variable in a class, if I want that variable directly accessible to subclasses, but not available to all classes, then which visibility modifier should I use?
- (b) (2 points) What is the difference between a checked and unchecked exception?

2. (3 points) **Code From Picture:** Consider the following picture:



Assuming that the `Person` class exists and has a constructor that takes `name` and `age` as arguments, write the appropriate lines of code to create the correct variables and objects that match what you see pictured above. Your code does not need to be inside a method or a class. (You are writing a short code snippet.)

3. (7 points) **Code Tracing:** Consider the following three classes, and indicate what is printed when the main method of the B class is executed. Place your answer (and nothing else) in the box under the code.

```
public class A extends B {
    @Override
    public void m(int a) {
        System.out.println("Hmm");
    }

    @Override
    public String toString() {
        return "A: " + super.toString();
    }
}

public class B {
    private String t;
    private int c;

    public B() {
        this.t = "121";
        this.c = 0;
    }

    public B(int a, int b) {
        this();
        System.out.println("Go " + a + " " + b);
        this.c = (a + b) / a;
    }

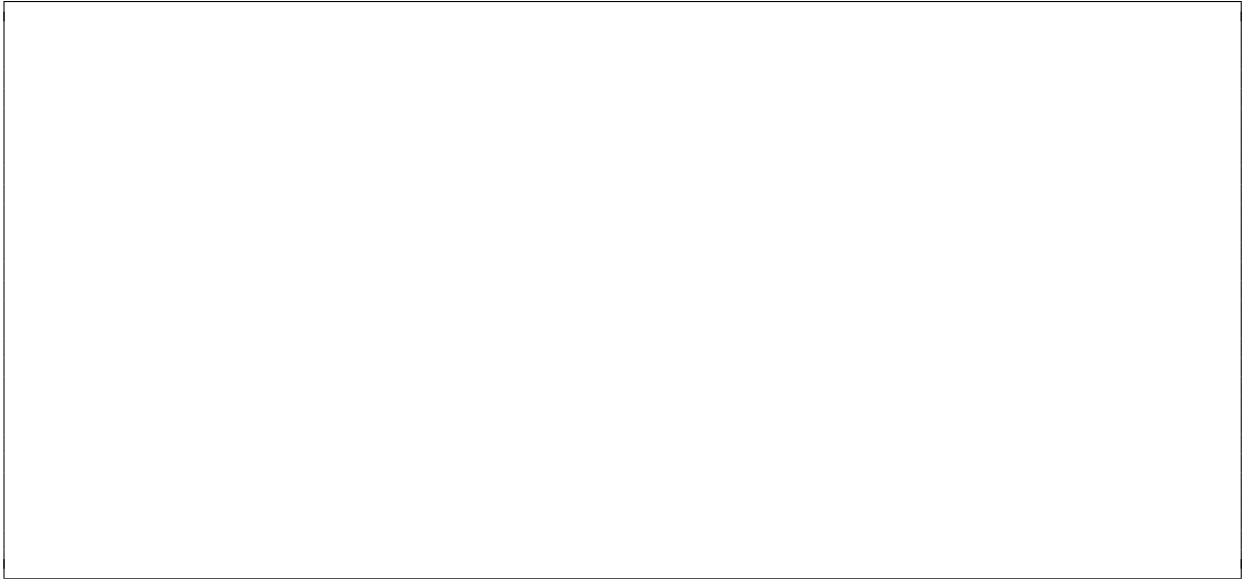
    public void m(int a) {
        for (int i = 0; i < a; i++) {
            this.t += "rocks";
        }
    }

    public String toString() {
        this.m(c);
        return t;
    }

    public static void helper(boolean one, boolean two, boolean three) {
        System.out.println(one + " " + two + " " + three);
    }

    public static void main(String[] args) {
        A ahmed = new A();
        B bill = new B(5, 10);
        C claire = new C(12, 4);
        helper(ahmed instanceof B, bill instanceof B, claire instanceof B);
        System.out.println(ahmed);
        System.out.println(bill);
        System.out.println(claire);
    }
}
```

```
public class C extends B {  
    private double d;  
  
    public C(int b, int a) {  
        super(a, b);  
        d = b;  
        d = d / a;  
    }  
  
    public String toString() {  
        return "C: " + super.toString();  
    }  
}
```



4. (26 points) **Free Response:** Computer Science Publications

In the field of Computer Science, researchers frequently disseminate their work through publications. There are two, main types of publications in Computer Science: Journal papers and conference papers. Imagine that you have a text file containing information about Computer Science publications. Each publication gets a single line, and the fields are separated by commas:

```
conference,Paper Title 1,Conference Name,Author 1, Author 2, Author 3
journal,Paper Title 2,Journal Name,Author 1, Author 2
```

Here is an example of a four line file containing information about four publications:

```
conference,Analog of Fagin's Theorem for Small Nondeterministic Finite Automata,Developments in
→ Language Theory,Christos A. Kapoutsis,Nans Lefebvre
journal,A Logical Characterization of Small 2NFAs,International Journal of Foundations of Computer
→ Science,Christos A. Kapoutsis,Lamana Mulafter
conference,BranchScope: A New Side-Channel Attack on Directional Branch Predictor,ASPLOS,Dmitry
→ Evttyushkin,Ryan Riley,Nael B. Abu-Ghazaleh,Dmitry Ponomarev
journal,Flexible Hardware-Managed Isolated Execution: Architecture Software Support and
→ Applications,IEEE Transactions on Dependable and Secure Computing,Dmitry Evttyushkin,Jesse
→ Elwell,Meltem Ozsoy,Dmitry Ponomarev,Nael B. Abu-Ghazaleh,Ryan Riley
```

(Note that in the file, each publication is on its own line. However, for the purposes of printing on this exam, each publication is printed across multiple lines to make it readable.)

On the pages that follow, fill in the code for the classes so that the supplied main function prints the output below.

```
public static void main(String[] args) {
    PublicationReader pr = new PublicationReader("pubs.txt");
    Publication[] arr = pr.getAllPubs();
    for (Publication p : arr) {
        System.out.println(p);
        System.out.println("----");
    }
    System.out.println("====");
    for (JournalPaper jp : pr.getJournalPubs()) {
        System.out.println(jp);
        System.out.println("----");
    }
}
```

Analog of Fagin's Theorem for Small Nondeterministic Finite Automata

By: Christos A. Kapoutsis, Nans Lefebvre

In Proceedings of: Developments in Language Theory

A Logical Characterization of Small 2NFAs

By: Christos A. Kapoutsis, Lamana Mulafter

Published in Journal: International Journal of Foundations of Computer Science

BranchScope: A New Side-Channel Attack on Directional Branch Predictor

By: Dmitry Evttyushkin, Ryan Riley, Nael B. Abu-Ghazaleh, Dmitry Ponomarev

In Proceedings of: ASPLOS

Flexible Hardware-Managed Isolated Execution: Architecture Software Support and Applications

By: Dmitry Evttyushkin, Jesse Elwell, Meltem Ozsoy, Dmitry Ponomarev, Nael B. Abu-Ghazaleh, Ryan Riley

Published in Journal: IEEE Transactions on Dependable and Secure Computing

====

A Logical Characterization of Small 2NFAs

By: Christos A. Kapoutsis, Lamana Mulafter

Published in Journal: International Journal of Foundations of Computer Science

Flexible Hardware-Managed Isolated Execution: Architecture Software Support and Applications

By: Dmitry Evttyushkin, Jesse Elwell, Meltem Ozsoy, Dmitry Ponomarev, Nael B. Abu-Ghazaleh, Ryan Riley

Published in Journal: IEEE Transactions on Dependable and Secure Computing

```

public class Publication {
    // This class contains two instance variables. You may not add any more.
    private String title;
    private ArrayList<String> authors;

    /**
     * Create a new publication based on the title
     *
     * @param title The title of the publication.
     */
    public Publication(String title) {

    }

    /**
     * Add an author to the publication
     *
     * @param author The name of the author to add.
     */
    public void addAuthor(String author) {

    }

    /**
     * Return a string containing the publication formatted as follows:
     *
     * <Title>
     * By: <Author 1>, <Author 2>, ...
     */
    public String toString() {

    }

}
}

```

```

public class JournalPaper extends Publication {
    // You may add whatever instance variables you need, but do not needlessly
    // duplicate information already stored in Publication


    public JournalPaper(String title, String journalName) {


    }

    public String toString() {


    }
}

public class ConferencePaper extends Publication {
    // You may add whatever instance variables you need, but do not needlessly
    // duplicate information already stored in Publication


    public ConferencePaper(String title, String conferenceTitle) {


    }

    public String toString() {


    }
}

```

```

public class PublicationReader {
    // This class contains only one instance variable. You may not add any more.
    private ArrayList<Publication> pubs = new ArrayList<Publication>();

    /**
     * Read the publications from the specified file and place them in pubs. When
     * reading in the publications, you must determine if it is a journal or
     * conference paper and create the appropriate object.
     *
     * @param filename The file containing the publications.
     */
    public PublicationReader(String filename) {

    }
}

```

The PublicationReader class continues on the next page.

```

/**
 * Return an array containing all of the publications.
 *
 * @return The array of publications.
 */
public Publication[] getAllPubs() {
    // Convert the ArrayList to an array and return it.
    return pubs.toArray(new Publication[0]);
}

/**
 * Return an array containing all of the journal publications.
 *
 * @return The array of journal publications
 */
public JournalPaper[] getJournalPubs() {

}

}

public static void main(String[] args) {
    PublicationReader pr = new PublicationReader("pubs.txt");
    Publication[] arr = pr.getAllPubs();
    for (Publication p : arr) {
        System.out.println(p);
        System.out.println("----");
    }
    System.out.println("====");
    for (JournalPaper jp : pr.getJournalPubs()) {
        System.out.println(jp);
        System.out.println("----");
    }
}
}

```