Name: _____ Andrew Id: _____

15-121 Fall 2025 Assessment 2

Up to 60 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.

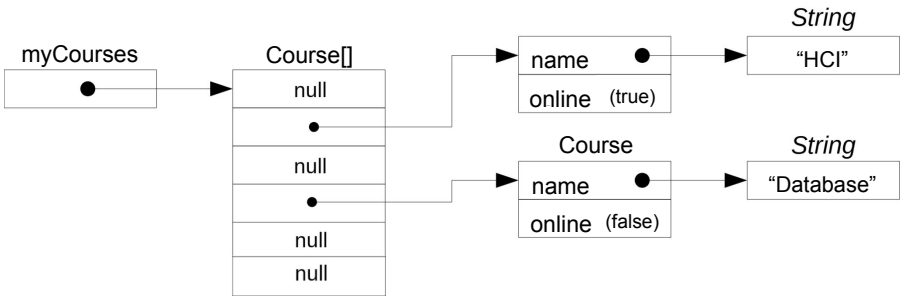
   (a) (2 points) Give one example of a checked exception and one example of an unchecked exception in Java.

   (b) (2 points) What is the difference between abstract data type and a data structure?

   (c) (3 points) Assuming each of the below classes is in an appropriately named file, does the following code compile? If not, why?

```
1
2  public class Person {
3      private String name;
4
5      public Person(String name) {
6          this.name = name;
7      }
8
9      public static void main(String[] args) {
10         Person p = new Student("John Smith", "InfoSys");
11         System.out.println(p.getMajor());
12
13     }
14 }
15
16 public class Student extends Person {
17     private String major;
18
19     public Student(String name, String major) {
20         super(name);
21         this.major = major;
22     }
23
24     public String getMajor() {
25         return this.major;
26     }
27 }
```

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

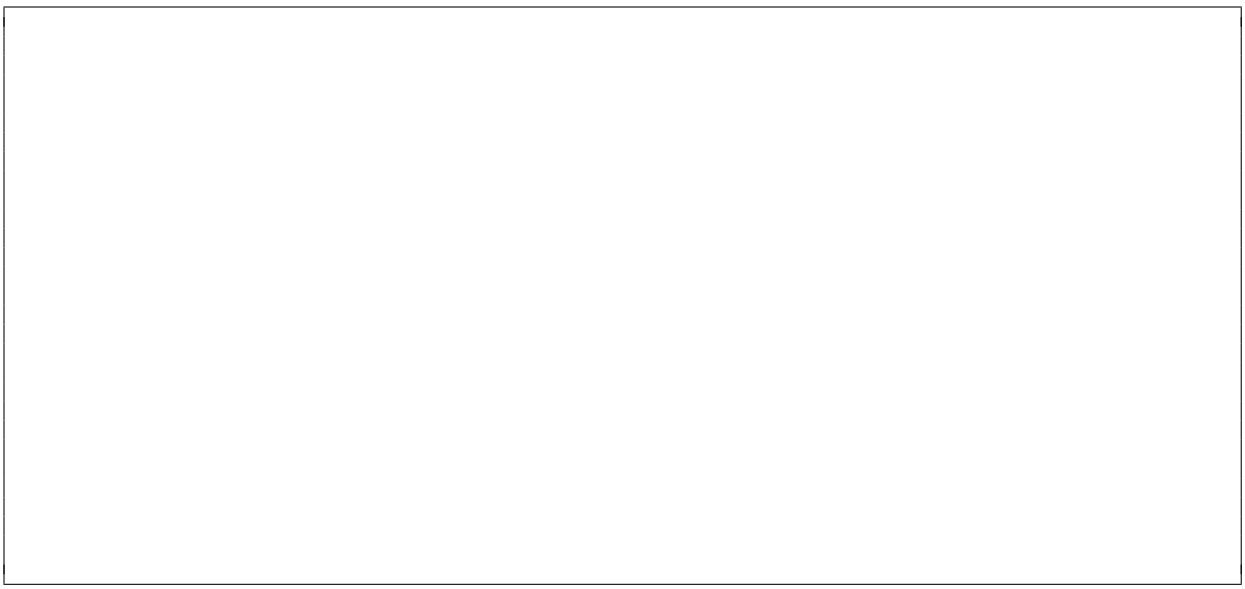


Assuming that the `Course` class exists and has a constructor that takes `name` and `online` as arguments, write the appropriate lines of code to create the correct variables and objects that match what you see pictured above. Your code does not need to be inside a method or a class. (You are writing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

3. (10 points) **Code Tracing**: Consider the following three classes, and indicate what is printed when the main method of the `A` class is executed. Place your answer (and nothing else) in the answer box on the next page.

```
public class A {
    protected String name;
    protected int x;

    public A() {
        System.out.println("ADC");
        init();
    }

    public A(String name) {
        this.name = name;
        System.out.println("A created: " + name);
    }

    public void init() {
        System.out.println("A.init");
    }

    public void action() {
        System.out.println("A.action: " + name);
    }

    public String getType() {
        return "A";
    }

    public int getX() {
        return this.x;
    }

    public static void main(String[] args) {
        A a = new A();
        A b = new B("Hello");
        C c = new C();
        b.action();
        c.action();
        System.out.println("Type: " + c.getType());
        if (c instanceof A) {
            System.out.println("x = " + ((A) c).getX());
        }
    }
}
```

```
public class B extends A {
    public B(String name) {
        super(name);
        this.x += 2;
        System.out.println("BC");
    }

    public void action() {
        super.action();
        System.out.println("B.action: " + name);
    }

    public String getType() {
        return "B";
    }

    public String toString() {
        return "B(" + super.toString() + ")";

    }
}

public class C extends A {
    @Override
    public void action() {
        this.x += 5;
    }

    public int getX() {
        return super.getX() * 2;
    }

    @Override
    public String getType() {
        return "C";
    }
}
```



4. (30 points) **Free Response:** Library Books

Libraries classify books into two types: **Fiction** and **Non-Fiction**. Imagine that you have a text file containing information about books. Each book gets a single line in the input file with the following fields, separated by commas:

- The type of the book (`fiction` or `nonfiction`)
- The genre (for fiction books) or the category (for nonfiction books)
- The book title
- The year of publication
- One or more authors

For example, here is a file with four books:

```
fiction,Fantasy,The Hobbit,1937,J.R.R. Tolkien
nonfiction,Memoir,Educated,2018,Tara Westover
fiction,Comedy,Good Omens,1990,Terry Pratchett,Neil Gaiman
nonfiction,History,Sapiens,2011,Yuval Noah Harari
```

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

```
public static void main(String[] args) {
    BookReader br = new BookReader("books.txt");
    Book[] arr = br.getAllBooks();
    for (Book b : arr) {
        System.out.println(b);
        System.out.println("----");
    }
    System.out.println("====");
    for (NonFiction nf : br.getNonFictionBooks()) {
        System.out.println(nf);
        System.out.println("----");
    }
}
```

Expected Output:

```
The Hobbit (1937)
By: J.R.R. Tolkien
Genre: Fantasy
----
Educated (2018)
By: Tara Westover
Category: Memoir
----
Good Omens (1990)
By: Terry Pratchett, Neil Gaiman
Genre: Comedy
----
Sapiens (2011)
By: Yuval Noah Harari
Category: History
----
====
Educated (2018)
By: Tara Westover
Category: Memoir
----
Sapiens (2011)
By: Yuval Noah Harari
Category: History
----
```

```java
public class Book {

    // Common attributes for all books
    private String title;
    private int year;
    private ArrayList<String> authors;

    /**
     * Construct a new Book with the given title and year.
     *
     * @param title The title of the book.
     * @param year  The year of publication.
     */
    public Book(String title, int year) {









    }

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







    }

    /**
     * Return a string containing the book formatted as follows:
     *
     * <Title> (<Year>)
     * By: <Author1>, <Author2>, ...
     */
    public String toString() {
```

```java
    }
}
```

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

    public Fiction(String title, int year, String genre) {

    }

    public String toString() {

    }
}
```

```java
public class NonFiction extends Book {
    // You may add whatever instance variables you need, but do not needlessly
    // duplicate information already stored in Book

    public NonFiction(String title, int year, String category) {

    }

     public String toString() {

    }
}
```

```
public class BookReader {
    // This class contains only one instance variable.
    // You may not add any more.
    private ArrayList<Book> books = new ArrayList<Book>();

    /**
     * Read the books from the specified file and place them in the books list.
     * When reading in the data, determine if it is Fiction or NonFiction
     * and create the appropriate object.
     * Authors should be added using addAuthor().
     *
     * @param filename The file containing the books.
     */
    public BookReader(String filename) {









    }
```

```
    /**
     * Return an array containing all of the books.
     *
     * @return The array of books.
     */
    public Book[] getAllBooks() {









    }

    /**
     * Return an array containing all of the non-fiction books.
     *
     * @return The array of non-fiction books.
     */
    public NonFiction[] getNonFictionBooks() {

















    }

    public static void main(String[] args) {
        BookReader br = new BookReader("books.txt");
        Book[] arr = br.getAllBooks();
        for (Book b : arr) {
            System.out.println(b);
            System.out.println("----");
        }
        System.out.println("====");
        for (NonFiction nf : br.getNonFictionBooks()) {
            System.out.println(nf);
            System.out.println("----");
        }
    }

}
```