Exam #2 Practice 3

VOCAB KEY

1 Exception

2 Thread

3 OOP

4 Process

5 Multiple Inheritance

6 Branch

7 Algorithm

8 Reentrant

9 Stack

10 Generic

MULTIPLE CHOICE KEY

1 D	6 B	11 A
2 D	7 C	12 D
3 B	8 C	13 B
4 D	9 B	14 A
5 A	10 A	15 C

FREE RESPONSE

1a.

```
@Override
public int hashCode() {
    return Objects.hash(title, isbn, weight);
}
```

1b.

```
@Override
public boolean equals(Object o) {
    if(o == this) return true;
    if(o == null || o.getClass() != this.getClass()) return false;
    Paperback pb = (Paperback) o;
    return pb.title.equals(title)
        && pb.isbn == isbn
        && pb.weight == weight;
}
```

1c.

```
private HashMap<Book, Double> priceList = new HashMap<>();
```

1d.

```
public void addBook(Book book, double price) {
    priceList.put(book, price);
}
```

1e.

```
@Override
public String toString() {
    String catalog = "";
    Iterator<Book> it = priceList.keySet().iterator();
    while(it.hasNext()) {
        Book b = it.next();
        catalog += String.format("%-50s $%6.2f\n", b, priceList.get(b));
    }
    return catalog;
}
```

1f.

```
public eBook(BufferedReader br) throws IOException {
     super(br);
     this.kilobytes = Integer.parseInt(br.readLine());
 }
```

1g.

```
public static void main(String[] args) {
    Bookstore store = null;
    try(BufferedReader br = new BufferedReader(new FileReader("test.books"))) {
        store = new Bookstore(br);
    } catch(IOException e) {
        System.err.println("Failed to read test.books: " + e);
        System.exit(-2);
    }
    System.out.println(store);
}
```

2a.

```
private static Object mutex = new Object();
private void downloadBook(String url) {
    String filename = download(url);
    synchronized(mutex) {
        files.add(filename);
    }
}
```

2b.

```
public void downloadBooks(String[] urls) {
    Thread[] threads = new Thread[urls.length];
    for(int i=0; i<threads.length; ++i) {
        final int j = i;
        threads[j] = new Thread(() -> downloadBook(urls[j]));
        // OR...
        // final String url = urls[j];
        // threads[j] = new Thread(() -> downloadBook(url));
        threads[j].start();
    }
    for(int i=0; i<threads.length; ++i) {
        try {
            threads[i].join();
        } catch(InterruptedException e) {
        }
    }
}
```

3a.

```
public static void print(Collection<?> c) {
    for(var v : c) System.out.println(v);
}

// OR...

public static <T> void print(Collection<T> c) {
    for(T v : c) System.out.println(v);
}
```

3b.

```
Iterator<Double> it = squares.iterator();
while(it.hasNext()) {
    Double d = it.next();
    if((Math.sqrt(d) % 1) != 0) it.remove();
}
```

Bonus

Use an iterator or a for-each loop.