

CSC 1351: Exam 1: Arrays and Inheritance

VERSION B

Name: _____

1 Arrays (Ch 6)

1.1

Does the following code compile? If it does not, how can it be fixed? If it does, what is its output? Does it throw an exception? If so, how can it be fixed?

```
import java.util.Arrays;
public class ReverseShift {
    // Shift values to the right
    static void shift(int[] s) {
        int tmp = s[s.length-1];
        for(int i=0;i<s.length-1;i++)
            s[i+1] = s[i];
        s[0] = tmp;
    }
    public static void main(String[] args) {
        int[] n = new int[3];
        for(int i=0;i<n.length;i++) n[i] = i;
        shift(n);
        for(int i=0;i<n.length-1;i++)
            for(int j=i+1;j<n.length;j++)
                assert(n[i] != n[j]);
        System.out.println(Arrays.toString(n));
    }
}
```

The code compiles, but when it runs it throws a `AssertionError`

replace the code: `for(int i=0;i<s.length-1;i++)`

with the code: `for(int i=s.length-2;i>=0;i--)`

1.2

Does the following code compile? If it does not, how can it be fixed? If it does, what is its output? Does it throw an exception? If so, how can it be fixed?

```
import java.util.*;
public class Shift {
    // shift values to the left
    static void shift(List<Integer> s) {
        int tmp = s.get(0);
        for(int i=1;i<s.size();i++)
            s.set(i-1,s.get(i));
        s.set(s.size()-1,tmp);
    }
    public static void main(String[] args) {
        List<Integer> n = new ArrayList<>();
    }
}
```

```

    for(int i=0;i<3;i++) n.add(i);
    shift(n);
    for(int i=0;i<n.size()-1;i++)
        for(int j=i+1;j<n.size();j++)
            assert(n.get(i) != n.get(j));
    System.out.println(n);
}
}

```

Compiles and runs

```
[1, 2, 0]
```

1.3

From the book:

Write a method: `public static boolean equals(int[] a,int[] b)` which compares two arrays to see if they are equal. If both are null, it should return true. If one is null and the other is not, it should return false. If the arrays are of different length, it should return false.

```

public class Equals {
    public static boolean equals(int[] a,int[] b) {
        if(a == null && b == null) return true;
        if(a == null || b == null) return false;
        if(a.length != b.length) return false;
        for(int i=0;i<a.length;i++)
            if(a[i] != b[i]) return false;
        return true;
    }
    public static void main(String[] args) {
        assert equals(null,null);
        assert !equals(new int[0],null);
        assert !equals(null,new int[0]);
        assert equals(new int[0],new int[0]);
        assert equals(new int[1],new int[1]);
    }
}

```

1.4

From Codingbat: Return true if the list contains, somewhere, three increasing adjacent numbers like 4, 5, 6, ... or 23, 24, 25.

```

tripleUp([1, 4, 5, 6, 2]) -> true
tripleUp([1, 2, 3]) -> true
tripleUp([1, 2, 4]) -> false

```

The function prototype is this: `boolean tripleUp(List<Integer> list) ...`

```

public static boolean tripleUp(List<Integer> array) {
    for(int i=0;i+2<array.size();i++) {
        if(array.get(i)+1 == array.get(i+1) && array.get(i+1)+1 == array.get(i+2))
            return true;
    }
    return false;
}

```

1.5

Does the following code compile? If it does not, how can it be fixed? If it does, what is its output? Does it throw an exception? If so, how can it be fixed?

```
public class A {
    int[] data = new int[10];
    public static void main(String[] args) {
        int c=0;
        for(int i=0;i<data.length;i++) {
            if(data[i] == 0)
                c++;
        }
        System.out.println("c="+c);
    }
}
```

The code does not compile.

replace the code: `int[] data = new int[10];`

with the code: `static int[] data = new int[10];`

2 Classes and Inheritance (Ch. 8/9)

2.1

Does the following code compile? If it does not, how can it be fixed? If it does, what is its output? Does it throw an exception? If so, how can it be fixed?

```
class Rocket {
    int energy = 3;
    void launch() { energy -= 2; check(); }
    void check() {
        if(energy <= 0) System.out.println("out_of_power");
        else if(energy <= 1) System.out.println("low_power");
    }
}

public class Starship extends Rocket {
    void launch() { energy -= 5; check(); }
    public static void main(String[] args) {
        Rocket r1 = new Rocket();
        Rocket r2 = new Starship();
        r1.launch(); r2.launch();
    }
}
```

Compiles and runs

low power
out of power

2.2

Does the following code compile? If it does not, how can it be fixed? If it does, what is its output? Does it throw an exception? If so, how can it be fixed?

```
public class Z {
    static int z;
```

```

int y;
public String toString() { return "("+z+","+y+")"; }
public static void main(String[] args) {
    Z z = new Z();
    z.z = 9; z.y = 3;
    Z q = new Z();
    q.z = 7; q.y = 8;
    System.out.println(z+" "+q);
}
}

```

Compiles and runs

(7,3) (7,8)

2.3

Does the following code compile? If it does not, how can it be fixed? If it does, what is its output? Does it throw an exception? If so, how can it be fixed?

```

class A {
    boolean isSuper() { return true; }
    void show() {
        if(isSuper()) System.out.println("Super!");
        else System.out.println("Not super.");
    }
}
public class B extends A {
    boolean isSuper() { return false; }
    public static void main(String[] args) {
        A a = new A();
        a.show();
        a = new B();
        a.show();
    }
}

```

Compiles and runs

Super!
Not super.

2.4

Does the following code compile? If it does not, how can it be fixed? If it does, what is its output? Does it throw an exception? If so, how can it be fixed?

```

class A { }
public class B extends A {
    public static void main(String[] args) {
        B a = (B)new A();
        System.out.println("works");
    }
}

```

The code compiles, but when it runs it throws a `ClassCastException`

replace the code: `B a = (B)new A();`

with the code: `A a = (B)new B();`

3 Anonymouse Inner Classes

3.1

Given the interface defined like this:

```
interface Adder {  
    void add(int a,int b);  
}
```

Write a complete program that uses this interface to print the sum of a and b to the screen.

```
interface Adder {  
    void add(int a,int b);  
}  
public class Test {  
    public static void main(String[] args) {  
        Adder adder = new Adder() {  
            public void add(int a,int b) {  
                System.out.println(a+b);  
            }  
        };  
        adder.add(3,5);  
    }  
}
```