1 Classes and Inheritance

1.1
Given the interface defined like this:

```java
public interface KeyListener {
    void keyPressed(char key);
    void keyReleased(char key);
}
```

Write a complete program that uses an anonymous inner class that implements this interface, then calls keyPressed() to print the key character ‘A’ to the screen.

```java
public class Test {
    public static void main(String[] args) {
        KeyListener kl = new KeyListener() {
            public void keyPressed(char c) {
                System.out.println(c);
            }
            public void keyReleased(char c) {
                System.out.println(c);
            }
        };
        kl.keyPressed('A');
    }
}
```

1.2
Does the following code compile? If it does not, how can it be fixed? If it does, what is it’s output? Does it throw an exception? If so, how can it be fixed?

```java
class A {}
class B extends A {
    public static void main(String[] args) {
        A a = new B();
        if(a instanceof B) b = a;
    }
}
```

The code does not compile.

replace the code: 
```java
if(a instanceof B) b = a;
```

with the code:
```java
B b = (B)a;
```

1.3
Does the following code compile? If it does not, how can it be fixed? If it does, what is it’s output? Does it throw an exception? If so, how can it be fixed?

```java
class A {}
class B extends A {}
class C extends A,B {}
class Main {
    public static void main(String[] args) {
        A a = new C();
    }
}
```
B b = new C();

The code does not compile.

replace the code: class C extends A,B {}
with the code: class C extends B {}
String[] letters = {"O","C","F","N","A","L","S","H","V","Y"};
String result = letters[5] + letters[1];
result = letters[4] + result;
result += letters[0];
result = letters[2] + result + letters[3];
StringBuilder sb = new StringBuilder();
sb.append(result);
for(int i=7;i< letters.length;i++)
sb.append(letters[i]);
System.out.println(sb.toString());

FALCONHVY

2.2
Find the best match between the column on the left and the column on the right.

| (1) extends   | 5 | (a) has no method bodies |
| (2) casting   | 6 | (b) has size() method    |
| (3) boolean   | 2 | (c) converts one type to another |
| (4) String    | 4 | (d) has length() method  |
| (5) interface | 8 | (e) default superclass   |
| (6) ArrayList | 1 | (f) used to create a subclass |
| (7) built-in array | 7 | (g) has fixed length |
| (8) Object    | 3 | (h) a primitive type     |

2.3
Does the following code compile? If it does not, how can it be fixed? If it does, what is it’s output? Does it throw an exception? If so, how can it be fixed?

```java
public class MyList {
    Object[] o = new Object[10];
    int size = 0;
    void add(Object d) { o[size++] = d; }
    Object get(int n) { assert n < size; return o[n]; }
    public static void main(String[] args) {
        MyList ml = new MyList();
        ml.add(3); ml.add(2); ml.add(1);
        System.out.println(ml.get(3));
    }
}
```

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

replace the code: System.out.println(ml.get(3));
with the code: System.out.println(ml.get(2));

2.4
Does the following code compile? If it does not, how can it be fixed? If it does, what is it’s output? Does it throw an exception? If so, how can it be fixed?

```java
public class Alist {
    int[] data = new int[]{3,4,5,9,7}; int size=4;
    public String toString() {
        StringBuilder sb = new StringBuilder();
        sb.append('{');
```
for(int i=0;i<size;i++) {
    if(i > 0) sb.append(',');
    System.out.println("i="+i);
    sb.append(i);
}
sb.append('}');
return sb.toString();
}

public class Alist {
    public static void main(String[] args) {
        System.out.println(new Alist());
    }
}

2.5

Write a method that creates a reversed copy of an array. Fill in the missing code. The number of lines corresponds to the answer key. Your code may vary.

import java.util.Arrays;

public class Reverse {
    static int[] reverse(int[] input) {
        int[] output = new int[input.length];
        ---------------------
        int n = input.length-1;
        ---------------------
        for(int i=0;i<input.length;i++)
            ---------------------
                output[n--] = input[i];
            ---------------------
        return output;
        ---------------------
    }
    public static void main(String[] args) {
        int[] in = new int[]{3,8,2,4};
        int[] out = reverse(in);
        System.out.println(Arrays.toString(in));
        System.out.println(Arrays.toString(out));
    }
}

// output:
[3, 8, 2, 4]
[4, 2, 8, 3]