

# 1 Java

## 1.1 Formal & Actual Type

```
1 public class q1 {
2     public static void main(String [] args) {
3         Object a = new Object();
4         Object b = new String("a");
5         int hash = a.hashCode();
6         String str = b.toString();
7     }
8 }
```

1. What is the formal type of variable **a**?
2. What is the actual type of variable **b**?
3. Will line 5 compile? Why or Why not?  
*Hint: Look up the Object class in the Java API*
4. Will line 6 compile? Why or Why not?  
*Hint: Look up the Object class in the Java API*
5. Assuming that line 6 compiles and that both the Object class and the String class have their own implementation of **toString**, which version runs in line 6? The Object implementation or the String implementation?

## 1.2 Static

```
1 public class q2 {
2     int x;
3     static int y;
4     public static void m1 () {
5         x = 5;
6         y = 6;
7     }
8     public void m2() {
9     }
10    public static void main(String [] args) {
11        m1();
12        m2();
13    }
14 }
```

1. There is an illegal method call in **main**. Which line is illegal?
2. There is an illegal assignment in **m1** . Which line is illegal?
3. Which variables can be used in **m2**?

### 1.3 Scope

```
1 public class q3 {
2     int x = 5;
3     static int y = 6;
4     public static void main(String[] args) {
5         q3 instance = new q3();
6         int x = 7;
7         int y = 8;
8         instance.run(x);
9         System.out.println(x);
10        System.out.println(y);
11    }
12    public void run(int x) {
13        x = 9;
14    }
15 }
```

1. This code prints

7

8

Modify lines 9, 10 and 13 so that the printout is

9

6

### 1.4 Generics

```
1 public class q4 {
2     private static class Node {
3         Object content;
4         public Node(Object input) {
5             this.content = input;
6         }
7         public Object getContent() {
8             return this.content;
9         }
10    }
11    public static void main(String[] args) {
12        Node a = new Node("Ada");
13        Node b = new Node(5);
14    }
15 }
```

1. Modify the Node class so that `getContent` for `a` returns a String and `getContent` for `b` returns an Integer.

## **2 Data Structures**

### **2.1 Stack**

1. What does it mean to push an item onto the stack?
2. What does it mean to pop an item from the stack?
3. If 1,2,3 are pushed onto the stack, in what order will they be reversed?

### **2.2 Queue**

1. What four letter acronym is used to describe the order in which items are added to a queue?
2. What do the four letters mean?

### **2.3 Heap**

1. In a heap, is the parent of the node greater than or less than the node?
2. Binary trees are used to represent heaps. Describe the shape of binary tree a heap is trying to maintain.
3. Can any node be removed from the heap? If not, which node(s) can be removed from the heap?

### **2.4 Dictionary**

1. Dictionaries are a collection of Key-Value pairs. The pairs can be stored in the dictionary and then one (either the key or the value) can be looked up using the other. Are keys retrieved using values or are values retrieved using keys?
2. Dictionaries are usually implemented using a hash table for fast lookup times. How does hashing allow for a fast lookup time?

## **3 Algorithms**

### **3.1 Sorting**

1. If a list is sorted, what is the first item? What is the last item?
2. Name three sorting algorithms.
3. The fastest guaranteed sorting algorithm is MergeSort. What is its complexity?

### 3.2 Binary Search

1. The height of a tree grows as the number of nodes increases. If a tree has  $n$  nodes, what is its height?
2. Given your previous answer, what is the complexity of looking up a value in a complete binary search tree? *Hint: This is the same complexity of searching for an item in a sorted array.*

### 3.3 Recursion

1. What are the two basic components of all recursive functions?
2. Lists are recursive data structures. A list can be thought of being composed of smaller lists. What is another recursive data structure?

## 4 Because for some reason I kept mentioning it:

### 4.1 `main`

1. Why must `main` be static?