

CS 102: Practice Test Problems for Test 1

1. What is the output to the screen?

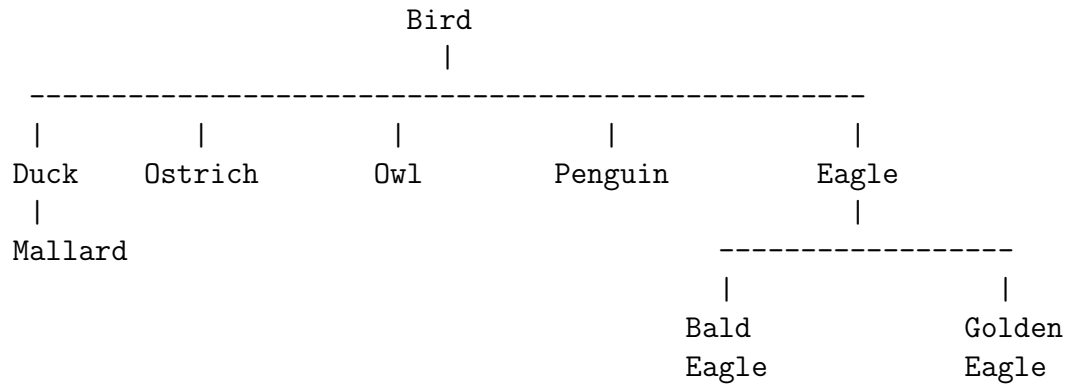
```
public class Question1{
    public static void main (String[] args){
        int[] a = {6,7,12,0,0,14};
        for (int i=0; i<a.length; i++)
            System.out.print(a[i] + " ");
        System.out.println();
        ch(a);
        for (int i=0; i<a.length; i++)
            System.out.print(a[i] + " ");
        System.out.println();
    }
    public static void ch (int[] ar){
        int temp = ar[0];
        for (int i=0; i<ar.length - 1; i++)
            ar[i] = ar[i+1];
        ar[ar.length - 1] = temp;
    }
}
```

Solution:

```
6 7 12 0 0 14
7 12 0 0 14 6
```

2. Put the classes Bird, Bald Eagle, Ostrich, Duck, Mallard, Golden Eagle, Owl, Eagle, Penguin in a class hierarchy.

Solution:



3. What is the output of the following code fragment?

```
String[] b = {"help","auto",argue"};
System.out.println(b.length + "\n" + b[2]);
```

Solution:

```
3
argue
```

4. Write a method

```
public static void printAsterisks(int[] ar)
```

which accepts the parameter `ar`, an array of integers, and prints out for each value stored in the array at index `index`, a line of `ar[index]` asterisks. For example, if `ar` holds the array `{3,4,5,0,1}`, the output to the screen should be

```
***
****
*****
*

```

Solution:

```
public static void printAsterisks(int[] ar){
    for (int val : ar){
        for (int i=0; i<val; i++){
            System.out.print('*');
            System.out.println();
        }
    }
}
```

5. (a) Write a method `product` which takes in two `int` parameters and returns their product.

Solution:

```
public static int product (int x, int y){
    return x*y;
}
```

- (b) Overload the method `product` to take in three `int` parameters and returns the product of all three parameters.

Solution:

```
public static int product (int x, int y, int z){
    return x*y*z;
}
```

6. Consider the following three classes P, Q, and R:

```
public class P{
    public String toString(){ return "Prince"; }
}
public class Q extends P{
    public String toString(){ return "Queen"; }
}
public class R extends P{
    public String toString(){ return "Royal"; }
}
```

What is the output to the screen of the following driver class?

```

public class PQRDriver{
    public static void main (String[] args){
        P[] ar = new P[3];
        ar[0] = new P(); ar[1] = new Q(); ar[2] = new R();
        for (int i=0; i<3; i++)
            System.out.println(ar[i]);
        }
    }
}

```

Solution:

```

Prince
Queen
Royal

```

7. What is *late binding* and how is this related to Java's implementation of polymorphism?

Solution: Late binding happens when the version of a method to be called from an object is determined not at compile time, but later at run time. A reference variable of a parent may for example refer to an object of one of several child classes, and so the method called from the reference variable decides at run time which version to call, depending on which child class the underlying object instantiates. This is an example of polymorphism, since the reference variable can refer to objects of different types at different points in time.

8. There is an error in the following lines of code. Rewrite the code to fix it. (Assume `ar` is an array.)

```

for (int i=1; i<=ar.length; i++)
    System.out.println(ar[i]);

```

Solution: The index `i` will be out of bounds when `i` is `ar.length` (since array indices run from 0 to `ar.length-1`). Presumably this code should print out all the elements of the array, and so should read something like

```

for (int i=0; i<ar.length; i++)
    System.out.println(ar[i]);

```

9. Assume there is a method

```
public static boolean isPrime (int n)
```

which returns `true` if the parameter `n` is a prime number, and which returns `false` otherwise.

Write a code fragment which declares an array `a` of 100 ints and stores in this array the first 100 prime numbers. (Call the method `isPrime` mentioned above; you do not have to implement the `isPrime` method itself. The first prime number is 2.)

Solution:

```
final int NUM_PRIMES = 100;
int[] a = new int[NUM_PRIMES];
int p = 2;
int count = 0;
while (count < NUM_PRIMES){
    if (isPrime(p)){
        a[count] = p;
        count++;
    }
    p++;
}
```

10. Consider the following two classes `Parent` and `Child`. There is an error concerning the visibility of data in the following code. Fix it.

```
public class Parent{
    private int n;
    public Parent(int num){ n=num; }
    public int getN(){ return n; }
}
public class Child extends Parent{
    public Child(int a){ super(a); }
    public double reciprocal(){ return 1./n;}
}
```

Solution: The error is that the class `Child` accesses the `private` data `n` of the class `Parent`. There are a few ways to fix this: One way is to make `n` `protected` instead of `private`, so that the `Child` class can access it. In this case, the second line would read

```
protected int n;
```

A better way would be to use the `getN()` method, since it is `public`. So in this case, the definition of the `reciprocal()` method would read

```
public double reciprocal(){ return 1./getN();}
```

11. Consider the following class `Num`, which we define as a user-defined wrapper class for an integer.

```
public class Num{
    private int n;
    public Num(int x){n=x;}
    public void setNum(int x){n=x;}
    public int getNum(){return n;}
    public String toString(){return n+"";}
}
```

- (a) Write a method (of a class other than `Num`)

```
public static void switchNums(Num n1, Num n2)
```

which switches the values of `n1` and `n2`.

Solution:

```
public static void switchNums(Num n1, Num n2){
    int temp = n1.getNum();
    n1.setNum(n2.getNum());
    n2.setNum(temp);
}
```

- (b) Is it possible to have a method

```
public static void switchInts(int x1, int x2)
```

which switches the values of the two `int` variables passed into the method? Explain your answer.

Solution: It is impossible to write such a method in Java. Primitive data such as `ints` are passed by value into methods. Therefore, even if the values of the parameters `x1` and `x2` are changed in the body of the method `switchInts`, the values of the variables passed into `x1` and `x2` will not be changed. (In fact, the solution to this problem is to wrap your `int` into a user-defined class such as `Num` above.)