Final Exam CIS 35a - Winter 2015

Question 1
Create a class called HasAProfile that maintains name, address and phone number information in serialized form on disk so that it can be retrieved whenever a new HasAProfile object is constructed. If an error occurs when retrieving the information during construction, the object will prompt the user for name, address and phone number information and use the information to create a new serialized profile on disk.

```
import java.io.*;
import java.util.*;
public class HasAProfile implements Serializable{
 private String name;
 private String address;
 private String phoneNo;

 HasAProfile (String name, String address, String phoneNo){
  this.name = name;
  this.address = address;
  this.phoneNo = phoneNo;
 }

public static void retrieve(String file){
 try {
  ObjectInputStream iob = new ObjectInputStream(new FileInputStream(file));
       try {
             HasAProfile profile = (HasAProfile) iob.readObject();
       } catch (ClassNotFoundException e)
       {
             System.out.println("Error”);
             Scanner scanner = new Scanner(System.in);
             String name = scanner.next();
             String address = scanner.next();
             String name = scanner.next();
             String address = scanner.next();
             String phoneNo = scanner.next();
             HasAProfile profile = new HasAProfile(name, address, phoneNo);
             ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(file));
             oos.writeObject(profile);
       }
 }
 catch (FileNotFoundException e)
 {
       System.out.println("File Not Found.");
 }
 catch (IOException e)
 {
       System.out.println("I O Error");
 }
}
}
```

Question 2
What will the following program print?

```
public class CompleteConcept {
      public static void main(String[] args) {
         String a = new String("Hello Universe!");
        System.out.println(a.toUpperCase());
        System.out.println(a.toLowerCase());
        System.out.println(a.length());
        System.out.println(a.charAt(0));
        System.out.println(a.indexOf('e'));
        System.out.println(a.indexOf("Uni"));
        System.out.println(a.substring(6));
        System.out.println(a.substring(6, 9));
        System.out.println(a.equals("hello universe!"));
        System.out.println(a.equalsIgnoreCase("hello universe!"));
         System.out.println(a.startsWith("Hello"));
        System.out.println(a.startsWith("Uni", 6));
        System.out.println(a.endsWith("e!"));
        System.out.println(a.contains("Uni"));
        System.out.println(a.replace('e', 'u'));
        System.out.println(a.replace("ll", "lll"));
        a = "   A B C  \n  ";
      System.out.println(a.trim());
    }
}
```

```
HELLO UNIVERSE!
hello universe!
15
H
1
6
Universe!
Uni
false
true
true
true
true
true
Hullo Univursu!
Helllo Universe!
A B C
```

Question 3
Identify errors in the following program, correct them and write the output.

```
class test {
                  public static void main(String[] args) {
                   byte a = 100;
                   short b = a * 3; //Datatype mismatch
                   long l = 2000;
                   float k = 284.24; //Datatype mismatch
                   byte c = k; //Datatype mismatch
```

```
                int m = a;
                double d = b;

                 System.out.println(b);
                System.out.println(c);
                System.out.println(d);
                }
            }
```

Question 4

File Encryption is the science of writing contents of a file in a secret code. Your encryption program should work like a filter, reading the content of one file, modifying the data into a code, and then writing the coded contents out to a second file. The second file will be a version of the first file, but written in secret code. Write a program to demonstrate the above working with binary files.

```
class FileEncryption {
      public static void main(String [] args)
      {
            int c = readAndCountStream("file1.dat");
            int readdata [] = new int[c];
            readstream("file1.dat", readdata);
            convert(readdata)
            writestream(file2.dat);

      }

      //Assuming a file of bytes -
      static void writeStream(String filename, int [] readata) {
             try {
                 FileOutputStream file = new
                     FileOutputStream(filename);
                 BufferedOutputStream buff = new
                     BufferedOutputStream(file);
                 for (int out = 0; out < readdata.length ; out++) {
                     buff.write(readdata[out]);
                 }
                 buff.close();
             } catch (IOException e) {
                 System.out.println("Exception: " + e.getMessage());
             }
         }

        static int readAndCountStream(String filename) {
            int ctr=0;
              try {
                 FileInputStream file = new
                     FileInputStream(filename);
                 BufferedInputStream buff = new
                     BufferedInputStream(file);
                 int in = 0;
                 do {
                     in = buff.read();
                   ctr++;
```

```
                    if (in != -1)
                        break;
                } while (in != -1);
                buff.close();
                return true;
            } catch (IOException e) {
                System.out.println("Exception: " + e.getMessage());
                return false;
            }
        }

    static void readstream(String filename, int c []) {
         int ctr=0;
           try {
                FileInputStream file = new
                    FileInputStream(filename);
                BufferedInputStream buff = new
                    BufferedInputStream(file);
                int in = 0;
                do {
                    c[in]= buff.read();
                  inr++;
                    if (in != -1)
                        break;
                } while (in != -1);
                buff.close();
            } catch (IOException e) {
                System.out.println("Exception: " + e.getMessage());
            }
        }


|
Question 5a
Generate the output of the following program:
    class Add {
         protected int i;
         Add(int a) {i = a;}
         protected void addIt(int amount) {i += amount;}
         protected int getIt() {return i;}
    }
     class DAdd extends Add {
         private int i;
         DAdd(int a, int b) {
               super(a);
               i = b;
         }
         protected void addIt(int amount) {i = i * super.i + amount;}
         protected int getIt() {return i + 1;}
         protected void doubleIt(int amount) {addIt(2 * amount);}
    }

    public class TestAdder {
```

```
        public static void main(String args[]) {
              Add A = new Add(3);
              DAdd DA = new DAdd(1, 5);
              A.addIt(2);
              System.out.println(A.getIt());
              A = DA;
              A.addIt(2);
              System.out.println(A.getIt());
              DA.doubleIt(2);
              System.out.println(A.getIt());
        }
  }

  Output
  5
  8
  12
```

Question 5b
What is the difference between method overriding and method overloading?
Overriding enables Polymorphism by setting up different behavior in methods with same signature in classes using extends (or Parent Child Relationship)
Overloading enables Polymorphism in a same class by setting up different behavior in methods with same signature.

When working with Interfaces? //Discussed in class.

Question 6

1. Why should you use Stringbuffer objects instead of String objects in a program that makes lot of changes to strings?
   StringBuffers reuse memory. Strings on other hand use new memory everytime a string value is changed. This is suboptimal performance wise.

2. Each of the numeric wrapper classes has a static toString method. What do these methods do?

This method converts a primitive (for which a wrapper class exists) into a String.

Question 7

1. What are some consideration for selecting a Collection?

Depends entirely on requirements. You can factor is size, usage (rate of inserts, updates versus selects), if data should be kept sorted or unsorted etc.

2. What is Garbage Collection and how can you mark an object for Garbage Collection.

Garbage Collection is reclaiming memory from usused elements (object, code or local variables). An object can be marked for Garbage Collection by setting it to null or by usage of finalize method.