Q1. Suppose we have a string String s = "red"; **What shall we get from** System.out.println(s == "red");**?**

Q2. this and **super**

(a) this - Read the following program. It can compile and run to give 5 lines of output.

(i) Please insert "`this.`" to the code as many as possible, so that the code still works in the same way.

(ii) Complete the blanks in the comments to show all outputs.

```
class A {
    public String tellMe() {return "a";}
    public void alg() {System.out.println(tellMe() + "*");}
}

class B extends A {
    public String tellMe() {return "b";}
    public void alg() {System.out.println(tellMe() + "#");}
}

class C extends A {
    public String tellMe() {return "c";}
}
```

```
public static void main(String [] args) {

    A x1 = new B(); x1.alg(); //b#

    B x2 = new B(); x2.alg(); //b#

    A x0 = new A(); x0.alg(); //_____________

    A x3 = new C(); x3.alg(); //_____________

    C x4 = new C(); x4.alg(); //_____________
}
```

(b) super - Now, "tellMe()" is changed to "toString()", and "super." Is added into the code. What will be the outputs?

[Recall - Lab07 Page 6] this and ***super*** work in very different way:

- When "this" is used in a statement: "this" means the runtime object (dynamic decision at runtime)
- When "super" is used in a statement: "super" means the parent class of that statement's class (confirm at compile time)

```
class A {
    public String toString() {return "a";}
    public void alg() {System.out.println(super.toString() + "*");}
}

class B extends A {
    public String toString() {return "b";}
    public void alg() {System.out.println(super.toString() + "#");}
}

class C extends A {
    public String toString() {return "c";}
}
```

```
public static void main(String [] args) {

    A x1 = new B(); x1.alg(); //a#

    B x2 = new B(); x2.alg(); //a#

    A x0 = new A(); x0.alg(); //A@2a139a55*  ("A@2a139a55" is obtained from .toString() of the Object class ^)

    A x3 = new C(); x3.alg(); //_____________

    C x4 = new C(); x4.alg(); //_____________
}
```

^https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html

Q3. Given the statements:

(i) `ArrayList<int> arrlist = new ArrayList<>();` //Compile-time error

(ii) `ArrayList<int[]> arrlist = new ArrayList<>();` //OK

The error message for (i) is: *Syntax error on token "int", Dimensions expected after this token*

Why does (i) contain error but (ii) is Okay? How to explain the error message?

Q4. [More on *for-each* loop] How to implement the following method?

Note: When a for-each loop iterates through a collection, we cannot add/delete elements. *Run-time error!!* ☠

```
public static void remove(ArrayList<Employee> allEmployees, Employee e) { //Remove e from allEmployees




}
```

Q5. [Lab07-Q7] How to solve?

```
public abstract class Account {
    private String account_no;
    public Account(String ano) {..}
    public String getAccountNumber() {..}
    ..
}
public class CreditCardAccount extends Account {
    private double balance; private double creditLimit;
    public CreditCardAccount(String ano, double b, double crl) {..}
    ..
}
public class SavingsAccount extends Account {
    private double balance;
    public SavingsAccount(String ano, double b) {..}
    ..
}
public class PowerAdvantageAccount extends Account {
    private SavingsAccount savingsAC;
    private CreditCardAccount creditCardAC;

    public PowerAdvantageAccount(String ano, SavingsAccount sa, CreditCardAccount cr) {..}
}
```

a1.txt

```
0123456789 5000
1111222222 6000
3333334444 1000
6666888888 8000 20000
9999123456 0123456789 6666888888
```

0... to 5...: Savings
6... to 8...: Credit Card
9... for PowerAantage

findAccount()

main()

```
public static void main(String [] args) throws FileNotFoundException {

    .. // Prepare scanner inFile for file reading


    while (inFile.hasNext()) {









    }
    infile.close();

    System.out.print("\nInput an account number to search: ");
    String ac=in.next();

    Account a=findAccount(acs,ac);
    if (a!=null)System.out.println("\n[Result]\n"+a.toString());
    else        System.out.println("\n[Result]\nThe account is not found.");

    in.close();
  }
```