Inheritance in Java

Question 1

Consider the following inheritance hierarchy:

```
class BaseCL {
    public BaseCL(){};
    public void demoFunc(){};

    private int m;

    protected int n;
}

class DerivedCL extends BaseCL {
    public DerivedCL(){};
    public void demoFunc(){};

    private int r;
}
```

(a) Which of the data members m, n, and r can be accessed by a member function in the derived class?

(b) Which of the data members m, n, and r can be accessed by a member function in the base class?

(c) Consider the test program:

```
class TestIn {
    public static void main(String[] args){
        BaseCL bObj = new BaseCL();
        BaseCL dObj = new DerivedCL();

        bObj.demoFunc();
        dObj.demoFunc();
    }
}
```

Which method does the method call `bObj.demoFunc()` invoke, and which method does the method call `dObj.demoFunc()` invoke?

Question 2

Write a generic class, named `Queue`, in Java for the queue type that uses a linked list to store the elements. The `Queue` class has a member variable, named `head`, that references the first node of the list, a member variable, named `tail`, that references the last node of the list, and a member variable, named `size`, that stores the number of elements in the queue. The `Queue` class provides the following methods: `push`, `pop`, `front`, and `empty`. The `Node` class is defined as follows.

```
class Node<T> {
    public T nodeValue;
    public Node<T> next;

    Node(T item){
        this(item, null);
    }

    Node(T item, Node<T> next){
        nodeValue = item;
        this.next = next;
    }
}
```

Write a class, named `DerivedQueue`, which extends `Queue` by providing a method named `emergency_push` that inserts an element at the front of the queue.

Question 3

Use the employee hierarchy (see below) and indicate the version of displayEmployeeInfo() that is executed by each of the method calls:

```
class TestEmployee {
    public static void main(String[] args){
        Employee p;

        SalaryEmployee q = new SalaryEmployee("Steve Howard","896-54-3217",3330.00);

        HourlyEmployee r = new HourlyEmployee("Johns Ross","896-54-3217",7.50,40);

        p = q;

        r.displayEmployeeInfo();
        q.displayEmployeeInfo();
        p.displayEmployeeInfo();
    }
}

// base class for all employees
abstract class Employee {
    public Employee(String name, String ssn){
        empName = name;
        empSSN = ssn;
    }

    // output basic employee information
    void displayEmployeeInfo() {
        System.out.println("Name: " + empName);
```

```
        System.out.println("Social Security Number:  " + empSSN);
    }

    // function with this prototype will exist in each derived class
    abstract void payrollCheck();

    protected String empName;
    protected String empSSN;
}

// salaried employee "is an" employee with a monthly salary
class SalaryEmployee extends Employee {
    // initialize Employee attributes and monthly salary
    public SalaryEmployee(String name, String ssn, double sal){
        super(name, ssn);
        salary = sal;
    }

    // update the monthly salary
    public void setSalary(double sal) {
        salary = sal;
    }

    // call displayEmployeeInfo from base class and add
    // information about the status (salaried) and weekly salary
    public void displayEmployeeInfo(){
        super.displayEmployeeInfo();
        System.out.println("Status:   salaried employee");
        System.out.println("Salary per week $" + salary);
    }

    // cut a payroll check with the employee name, social security
    // number in angle brackets, and salary
    void payrollCheck(){
        System.out.println("Pay " + empName + " (" + empSSN + ")  $" + salary);
    }

    // salary per pay period
    private double salary;
}

// hourly employee "is an" employee paid by the hour
class HourlyEmployee extends Employee {
    // initialize Employee attributes, hourly pay rate
    // and hours worked
    public HourlyEmployee(String name, String ssn, double hp, double hw){
        super(name,ssn);
        hourlyPay = hp;
```

```
        hoursWorked = hw;
    }

    // update the hourly pay and hours worked
    public void setHourlyPay(double hp){
        hourlyPay = hp;
    }

    public void setHoursWorked(double hw){
        hoursWorked = hw;
    }

    // call displayEmployeeInfo from base class and output info
    // on hourly rate and scheduled hours
    public void displayEmployeeInfo(){
        super.displayEmployeeInfo();
        System.out.println("Status:   hourly employee");
        System.out.println("Payrate:  $" + hourlyPay + " per hour");
        System.out.println("Work schedule (hours per week) " + hoursWorked);
    }

    public void payrollCheck(){
        System.out.println("Pay " + empName + " (" + empSSN + ")  $" + (hourlyPay * hours
    }

    // pay based on hourly pay and hours worked
    private  double hourlyPay;
    private double hoursWorked;
}
```