TK 1143 Program Design

Inheritance & Polymorphism

Part A

1. Identify Errors
   a) Identify **EIGHT** syntax errors in the following Java program segment and modify the program to remove the errors.

```
public class Student {
   private String name;
   protected String matricNum;

   public Student(String n, int m) {
      name = n;
      matricNum = m;
   }
   public void displayDetails() {
      System.out.println("Name: "+name);
      System.out.println("Matric #: "+matricNum);
   }
}
public class UnderGradStudent Student {
   private Course courses;

   public UnderGradStudent(String n, String m) {
      courses = new Course[3];
   }
}
public class PostGradStudent extends Students {
   private String supervisor;

   public PostGradStudent(String n, String m, String s) {
      supervisor = s;
   }

   public String displayDetails() {
      System.out.println("Name: "+name);
      System.out.println("Matric #: "+matricNum);
      System.out.println("Supervisor: "+supervisor);
   }
}
```

   b) What is wrong with the following code? Why it is showing compilation error?

```
public class A
{
    public A(){
        System.out.println(1);
        super();
        System.out.println(2);
    }
   }
```

2) What will be the output of the following program?

a) Topic : Constructor

```
public class A{
    public A(){
        System.out.println("Class A Constructor");
    }
}
public class B extends A {
    public B(){
        System.out.println("Class B Constructor");
    }
}
public class C extends B{
    public C() {
        System.out.println("Class C Constructor");
    }
}
public class TesterClass{
    public static void main(String[] args) {
        C c = new C();
    }
}
```

b) Topic : Overiding method Vs Overloading Method

```
public class SeaCreature {
    public void method1() {
         System.out.println("creature 1");
    }
    public void method2() {
         System.out.println("creature 2");
    }
    public String toString() {
         return "ocean dwelling";
    }
}
public class Mammal extends SeaCreature {
    public void method1() {
         System.out.println("warm blooded");
    }
}
public class Whale extends Mammal {
    public void method1() {
         System.out.println("spout");
    }
    public String toString() {
         return "BIG!";
    }
}
```

```
public class Squid extends SeaCreature {

     public void method2() {
           System.out.println("tentacles");
     }
     public String toString() {
           return "squid";
     }
}

public class TesterSeaCreature {
    public static void main (String []args){
          SeaCreature[] elements = {new Squid(), new Whale(),
                                new SeaCreature(),new Mammal()};

          for (int i = 0; i < elements.length; i++) {
                System.out.println(elements[i]);
                elements[i].method1();
                elements[i].method2();
                System.out.println();
          }
    }
}
```

c) Topic : Static method

```
class X{

    static void staticMethod(){

        System.out.println("Class X");

    }

}

class Y extends X{

    static void staticMethod(){

        System.out.println("Class Y");

    }

}

public class MainClass{

    public static void main(String[] args){

        Y.staticMethod();}

}
```

d) Topic : Super keyword and this

```
public class M{
    protected int i = 12;

    public M(int j){
        System.out.println(i);
        this.i = j *10;
    }
}

public class N extends M{
    public N(int j){
        super(j);
        System.out.println(i);
        this.i = j * 2;
    }
}

public class MainClass{
    public static void main(String[] args){
        N n = new N(25);
        System.out.println(n.i);
    }
}
```

3) Draw a diagram to represent the hierarchical relationship between Mammal, Goat, Reptile, Animal, Cow and Frog.

4) Based on the following class Officer, answer 4 (a) – (c)

```
class Officer {
   private String name;
   private String address;

   public Officer(String n, String a) {
      name = n;
      address = a;
   }
   public void displayInfo() {
      System.out.println("\nName        : " + name);
      System.out.println("Address     : " + address);
   }
}
```

a) Fill in the blank to complete class NormalOfficer. Given that NormalOfficer is an Officer.

```
public class NormalOfficer ______________________________ {
   int salaryDay;
   int workingDay;

   public NormalOfficer(String n, String a,  int salary, int day) {

      ______________________

      salaryDay = ________________ ;

      workingDay = _______________ ;
   }
   public void displayInfo() {

   //Display info name and address

      ______________________________

      System.out.println("Salary      : " + ________________ );

      System.out.println("Working Day : " + workingDay );
   }
}
```

b) Write the definition for class ExecOfficer that extends NormalOfficer. Add a new attribute named bonus, and output the value in displayInfo() method.

c) Create a tester class and name it as TesterOfficer. Create and object of ExecOfficer, pass value to the constructor and call appropriate method(s) to print the output as follows.

Sample Output:

```
Name        : Aimy Azzalea
Department  : Sains Komputer
Salary      : 4500
Working Day : 25
Bonus       : 1000
```

Part B

Problem Statement

Write the code to implement the concept of inheritance for Vehicles. You are required to implement inheritance between classes. You have to write four classes in java i.e. one superclass, two sub classes and one driver (tester) class.

`Vehicle` is the super class whereas `Bus` and `Truck` are sub classes of `Vehicle` class. `Transport` is a driver class, which contains main method.

Detailed description:

Detailed description of `Vehicle` (Super class):

The class `Vehicle` must have following attributes:

1. Vehicle model
2. Registration number
3. Vehicle speed (km/hour)
4. Fuel capacity (liters)
5. Fuel consumption (kilometers/liter)

The `Vehicle` class must have following methods:

1. Parameterized `constructor` that will initialize all the data members with the given values.

2. `Getters` and `Setters` for each data member that will get and set the values of data members of class.

3. A method `fuelNeeded()` that will take *distance (in kilometer)* as an argument. It will calculate the amount of fuel needed for the given distance and will print the value of fuel needed for given distance. You can use the attributes '*Fuel consumption'* defined within above Vehicle class to determine the fuel needed for the given distance. You are required to implement this functionality by yourself.

4. A method `distanceCovered()` that will take *time* (in hours) as an argument. It will calculate the distance for the given time and speed and print the value of distance. The formula to calculate speed is given as `speed = distance/time`. You can use this formula to calculate the distance.

5. A `display()` method that will display all the information of a vehicle.

Detailed description of `Truck` (Sub class):

- The class `Truck` must have following attribute:
    - Cargo weight limit (Kilograms)

The `Truck` class must have following methods:

1. Parameterized `constructor` that will initialize all data members with the given values.
2. `Getters` and `setters` for each data member that will get and set the values of data members of class.
3. It must also override the `display()` method of Vehicle class and must call `display()` method of superclass within overridden method.

Detailed description of `Bus` (Sub class):

- The class `Bus` must have following attribute:
    - No of passengers

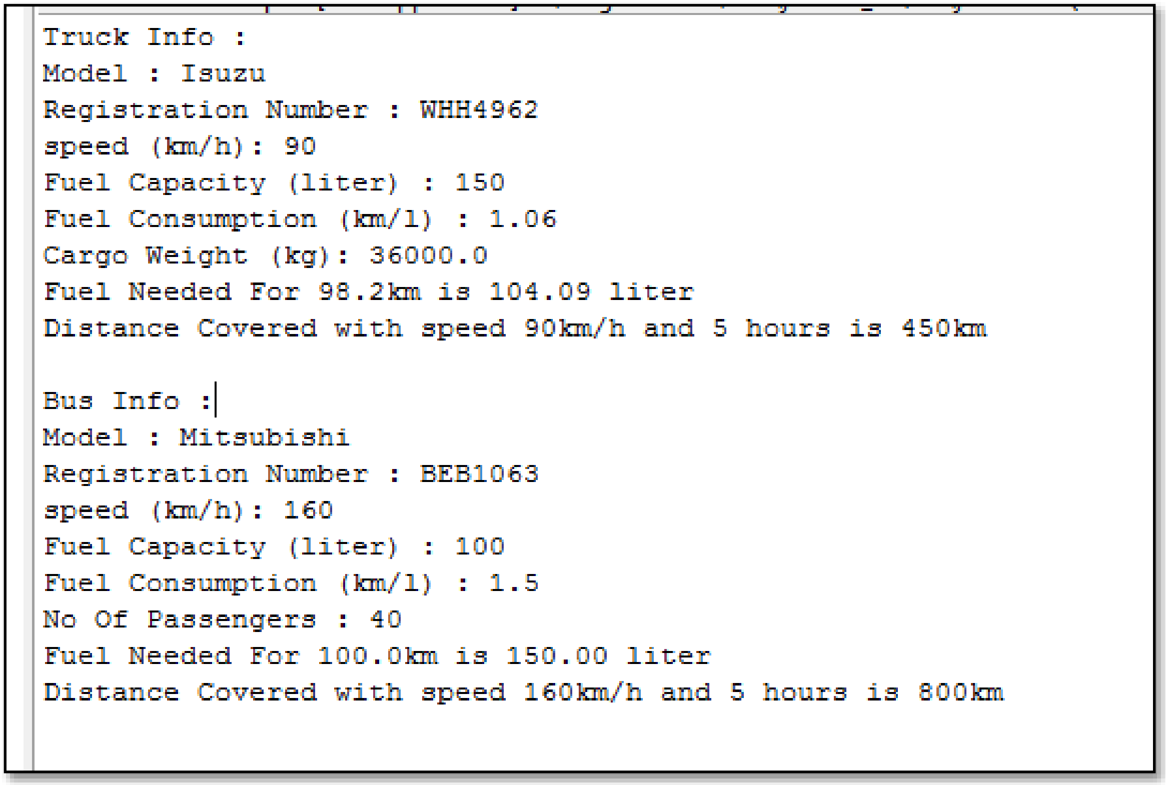
The `Bus` class must have following methods:

1. Parameterized `constructor` that will initialize all the data members with given values.
2. `Getters` and `setters` that will get and set the value of each data member of class.
3. It must also override the `display()` method of Vehicle class and must call `display()` method of super class within overridden method.

Create a class `Transport` which contains the main method. Perform the following within main method:

- Create an instance of class `Truck` and initialize all the data members with proper values.
- Create an instance of class `Bus` and initialize all the data members with proper values.
- Now, call `display(), fuelNeeded(),` and `distanceCovered()` methods using objects of these classes.

Example Output :



```
Truck Info :
Model : Isuzu
Registration Number : WHH4962
speed (km/h): 90
Fuel Capacity (liter) : 150
Fuel Consumption (km/l) : 1.06
Cargo Weight (kg): 36000.0
Fuel Needed For 98.2km is 104.09 liter
Distance Covered with speed 90km/h and 5 hours is 450km

Bus Info :
Model : Mitsubishi
Registration Number : BEB1063
speed (km/h): 160
Fuel Capacity (liter) : 100
Fuel Consumption (km/l) : 1.5
No Of Passengers : 40
Fuel Needed For 100.0km is 150.00 liter
Distance Covered with speed 160km/h and 5 hours is 800km
```