Inheritance:

1. Define a class named Payment that contains an instance variable of type double that stores the amount of the payment and appropriate accessor and mutator methods. Also create a method named paymentDetails that outputs an English sentence to describe the amount of the payment.

   Next, define a class named CashPayment that is derived from Payment. This class should redefine the paymentDetails method to indicate that the payment is in cash. Include appropriate constructor(s).

   Define a class named CreditCardPayment that derived from Payment. This class should contain instance variables for the name on the card, expiration date, and credit card number. Include appropriate constructor(s). Finally, redefine the paymentDetails method to include all credit card information in the printout.

   Create a main method that creates at least two CashPayment and two CreditCardPayment objects with different values and calls paymentDetails for each.

2. Create a class called Vehicle that has the manufacturer's name (type String), number of cylinders in the engine (type int), and owner (type Person given below). Then, create a class called Truck that is derived from Vehicle and has the following additional properties: the load capacity in tons (type double since it may contain fractional part) and towing capacity in RM (type int). Be sure your class has a reasonable complement of constructors, accessor and mutator methods, and suitably defined equals and toString methods. Write a program to test all your methods.

   The definition of the class Person is below. Completing the definitions of the methods is part of this exercise.

```
public class Person
{
        private String name;
        public Person()
        {…}
        public Person (String theName)
        {…}
        public String getName()
        {…}
        public void setName (String theName)
        {…}
        public String toString()
        {…}
        public Boolean equals(Object other)
        {…}
}
```

3. Define a class named Document that contains an instance variable of type String named text that stores any textual content for the document. Create a method named toString that returns the text filed and also include a method to set this value.

   Next, define a class for Email that is derived from Document and includes instance variables for the *sender*, *recipient*, and *title* of an email message. Implement appropriate accessor and mutator methods. The body of the email message should be stored in the inherited variable *text*. Redefine the toString method to concatenate all text field.

   Similarly, define a class for File that is derived from Document and includes a instance variable for the *pathname*. The textual contents of the file should be stored in the inherited variable *text*. Redefine the toString method to concatenate all text field.

   Finally, create several sample objects of type Email and File in your main method. Test you objects by passing them to the following subroutine that returns true if the object contains the specified keyword in the text property.

```
public static Boolean ContainKeyword (Document docObject, String keyword)
{
        if (docObject.toString().indexOf(keyword, 0) >= 0)
                return true;
        return false;
}
```