

CSC1016S Assignment 9

Inheritance and Polymorphism

Assignment Instructions

This assignment involves constructing programs in Java using sub-classes, inheritance and polymorphism.

Question 1 [25 marks]

Write a program called `Question1.java` to store some details of computer boxes (serial number, manufacturer, colour, amount of memory), screens (serial number, manufacturer, colour, size) and accessories (serial number, manufacturer, colour) in a single list, with the options of listing all parts or deleting a particular part.

Your program must continuously prompt the user to choose an option from a list/array and act on that option, until the exit option is chosen. See the sample output below. If a part is not found, output "Not found" instead of "Done".

You must use inheritance and polymorphism to model your Box, Screen and Accessory classes (use those exact class names) as subclasses of the same base class, which forms the basis for the list/array.

Note: Be very careful to reproduce the output exactly. Copy-and-paste is highly recommended to avoid minor typographical errors that drive you crazy when submitting online! However, pay attention to the differences in character encodings between PDF and your IDE.

Sample IO (The input from the user is shown in **bold font** – do not program this):

```
Welcome to Great International Technology
MENU: add (B)ox, add (S)creen, add (A)ccessories, (D)delete, (L)ist, (Q)uit
b
Enter the serial number
123
Enter the manufacturer
Lenovo
Enter the colour
Beige
Enter the amount of memory (MB)
2048
Done
MENU: add (B)ox, add (S)creen, add (A)ccessories, (D)delete, (L)ist, (Q)uit
s
Enter the serial number
456
Enter the manufacturer
Phillips
Enter the colour
Beige
```

Enter the screen size in inches

19

Done

MENU: add (B)ox, add (S)creen, add (A)ccessories, (D)eleate, (L)ist, (Q)uit

a

Enter the serial number

789

Enter the manufacturer

Logitech

Enter the colour

Beige

Done

MENU: add (B)ox, add (S)creen, add (A)ccessories, (D)eleate, (L)ist, (Q)uit

d

Enter the serial number

456

Done

MENU: add (B)ox, add (S)creen, add (A)ccessories, (D)eleate, (L)ist, (Q)uit

l

Box: 123, Lenovo, Beige, 2048

Accessories: 789, Logitech, Beige

Done

MENU: add (B)ox, add (S)creen, add (A)ccessories, (D)eleate, (L)ist, (Q)uit

q

Question 2 [35 marks]

Write a program to manage a set of graphical objects based on the commands in a specified file. This is a graphical equivalent of the famous **edlin** tool for text manipulation.

On the Amathuba page of the assignment, you will find `VectorGraphics` and `Question2` classes that will handle the file I/O for you. When you run `Question2`, specify the file name like this:

```
java Question2 myfile.in
```

Each line of the file contains an instruction in one of the formats given below:

```
a <id> <x> <y> rectangle <x_length> <y_length>
```

```
a <id> <x> <y> hline <x_length>
```

```
a <id> <x> <y> vline <y_length>
```

```
a <id> <x> <y> ptline <x1> <y1>
```

```
w
```

```
d <id>
```

```
m <id> <x> <y>
```

```
x
```

Explanation:

‘a’ adds an object of a particular type (rectangle/hline/vline/ptline) with the specified parameters.

‘w’ renders/draws the objects and prints the composite image to the screen.

‘d’ deletes an object based on its id.

'm' moves an object to a new position based on its id.

'x' exits the program.

Note:

- A 'ptline' differs from the other objects in that the values after the object type are not lengths, but instead a second point described by $x1$ and $y1$. This object represents a line from (x, y) to $(x1, y1)$.
- The answers the automatic marker expects were generated using an implementation of Bresenham's Line Drawing Algorithm. ([See the appendix](#)).
- Assume that the drawing canvas is 20 blocks wide (in the x direction) and 20 blocks high in the y direction), starting with (0,0) in the top left.

As mentioned before, you are given class that will serve as the framework for this program on the Amathuba page, including the main class that does the input and output. You are only required to create the **Rectangle**, **HLine**, **VLine** and **PtLine** subclasses of the given **VectorObject** base class. You should add comments to all classes you write.

Sample Input File

```
a 1 5 5 rectangle 10 10
a 2 3 3 hline 14
a 3 3 16 hline 14
a 4 3 3 vline 14
a 5 16 3 vline 14
w
d 1
m 2 3 5
m 3 3 14
m 4 5 3
m 5 14 3
a 6 6 6 ptline 13 13
a 6 13 6 ptline 6 13
w
```

Page 4 of 7

Question 3 [40 marks]

Part I [10 marks]

Create a class called Person. Assume that all persons have a name, age and gender. Create another class called Student derived from the Person class. A student has the following additional attributes, name of institution, programme of study, year of study and hobbies. Include constructors and accessor methods for these classes as appropriate.

Part II [30 marks]

This second part of the question makes use of some object(s) created in Part I above. Now, your task is to create two classes. You have to create a class called Car that is derived from Vehicle satisfying the following specifications:

class Vehicle

A vehicle object that has the following attributes: number of cylinders, name of the manufacturer and the owner (type Student).

Constructors

```
Vehicle (int numCylinders, String maker, Student owner)
```

```
    // create a new Vehicle object.
```

```
    // The vehicle owner is a Student
```

```
Vehicle(Vehicle v)
```

```
    // Copy constructor
```

Methods

```
public String toString()
```

```
    //print a string representation of the Vehicle object information.
```

class Car

A car object that inherits from a Vehicle object. A Car object has the following additional attributes: seating capacity and weight.

Constructors

```
Car(int numCylinders, String maker, Student owner, int passengers, double carWeight)
```

```
    //Create a new car object.
```

```
Car(Car C)
```

```
    // Copy constructor
```

Methods

```
@Override
```

```
public String toString()
```

```
    //print a string representation of the Car object information.
```

Last but not least, write a program called `Question3.java` which acts as your driver class to test your classes above. Your driver class, having the main method, must conform to the sample input output given below:

Sample IO (The input from the user is shown in **bold font** – do not program this):

Enter the vehicle manufacturer:

Land Rover

Enter the name of the vehicle owner:

Kennedy Monroe

Enter the owner's gender:

Male

Enter the owner's programme of study:

BSc in Engineering in Chemical Engineering

Enter the owner's Institution name:

University of Cape Town

Enter the owner's hobbies:

hiking

Enter the owner's age:

21

Enter the number of cylinders in the engine:

8

Enter the car seating capacity:

6

Enter the weight of the car:

25500

Land Rover, 8 cylinders, owned by Kennedy Monroe, a 21-year old Male studying BSc in Engineering in Chemical Engineering at University of Cape Town. Kennedy Monroe likes hiking.

The car is a 6-seater weighing 25500.0 kg

Marking and Submission

Submit your assignment as a set of Java files named Question1.java, Box.java, Screen.java, Accessory.java, Rectangle.java, HLine.java, VLine.java, PtLine.java, Person.java, Student.java, Vehice.java, Car.java and Question3.java containing classes with corresponding names, as well any other Java files you have created, in a Zip file. Use the naming convention: **Yourstudentnumber.zip**

Appendix

Pseudocode for line drawing algorithm

```
def draw( $x_0, y_0, x_1, y_1$ )=  
    steep =  $|y_1 - y_0| > |x_1 - x_0|$   
    if steep then  
        swap( $x_0, y_0$ )  
        swap( $x_1, y_1$ )  
    end if  
    if  $x_0 > x_1$  then  
        swap( $x_0, x_1$ )  
        swap( $y_0, y_1$ )  
    end if  
    if  $y_0 < y_1$  then  
        ys = 1  
    else  
        ys = -1  
     $m = |y_1 - y_0| / (x_1 - x_0)$   
     $y = y_0$   
    error = 0  
  
    for  $x = x_0$  to  $x_1$  do  
        if steep then  
            plot( $x, y$ )  
        else  
            plot( $y, x$ )  
        end if  
        error = error + m  
        if error > 0.5 then  
             $y = y + ys$   
            error = error - 1  
        end if  
    end for  
end
```

NOTE:

- This code does not deal with horizontal or vertical lines.
- $|a|$ represents the absolute value of a i.e. the value without regard to its sign.