Q1 Apply the Abstract Factory Design Pattern to answer the following questions.

(a) Brielfy describe the main purpose of Abstract Factory Design Pattern. [4 marks]

(b) Draw class diagrams to illustrate the Abstract Factory Design Pattern. Your diagrams should shown all necessary methods, associations between classes, inheritance(s), and note(s). [5.5 marks]

(c) Peter wants to develop a program which can create different kinds of robots by assembling different kinds of arm(s) and foot(s). Initially, he has written the basic *Robot*, *Arm*, and *Foot* classes. He has also written the *Production* class to perform the creation of a *Robot* object. The following are Java codes for the classes:

```
public class Arm {
  String name;

  public Arm(String name) { this.name = name;}
  public String toString() { return name; }
} // Arm

public class Foot {
  String name;

  public Foot(String name) {this.name = name; }
  public String toString() { return name; }
} // Foot

public class Robot {
  Arm rightArm, leftArm;
  Foot rightFoot, leftFoot;

  public Robot (Arm rm, Arm lm, Foot rf, Foot lf) {
    this.rightArm = rm; this.leftArm = lm;
    this.rightFoot = rf; this.leftFoot = lf;
  }

  public String toString() {
    return "Robot: " + rightArm.toString() + ", " +
    leftArm.toString() + ", " + rightFoot.toString() + ", " +
    leftFoot.toString();
  }
}  // Robot
```

Question Q1 to be continued on next page.

Question Q1 continues from previous page.

```
public class Production {
  public static void main (String[] args) {
    Robot aRobot;

    Arm sRtArm = new Arm("Right S Arm");
    Arm sLtArm = new Arm("Left S Arm");
    Foot sRtFoot = new Foot ("Right S Foot");
    Foot sLtFoot = new Foot ("Left S Foot");
    aRobot = new Robot (sRtArm, sLtArm, sRtFoot, sLtFoot);

    System.out.println("Robot created " + aRobot);
  }
} // Production
```

Peter shows his classes to his supervisor, Mary. Mary provides the *AdvancedArm* class, which is a subclass of *Arm*, and the *AdvancedFoot* class, which is a subclass of *Foot*, for creating an object of the *AdvancedRobot* class, which is a subclass of *Robot*. Peter has found that the *Production* class needs to be changed for creating an *AdvancedRobot* object with *AdvancedArm* objects and *AdvancedFoot* objects. You are asked to apply the Abstract Factory Design Pattern to re-design the above program so that the *Production* class can create a *Robot* object or an *AdvancedRobot* object by using a different factory class. Draw a class diagram to show the new design of the program. Show all the necessary methods, classes, interface(s) and association(s) in your class diagram. [8.5 marks]

(d) Write the Java codes for implementing the revised *Production* class and one of the concrete factory classes in your class diagram given in your answer to Q1 Part (c).

[7 marks]