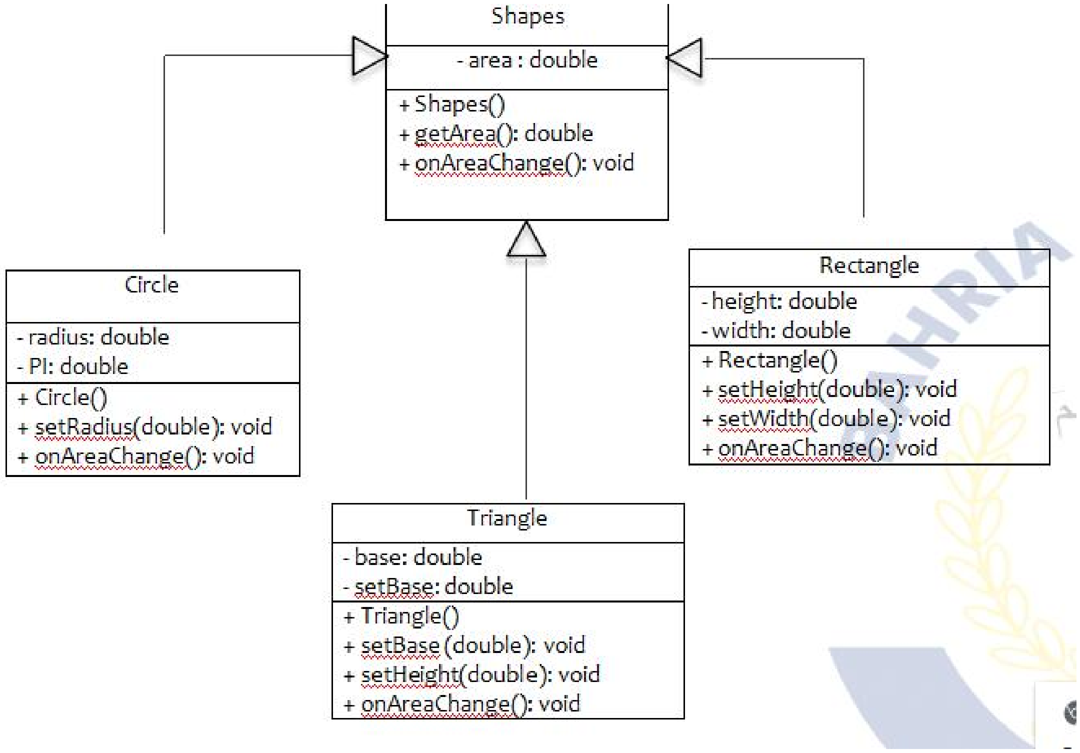
LAB # 07

1) **Task # 01:** Write the classes below containing the given instance variables and methods, following the inherited hierarchy:



Solution:

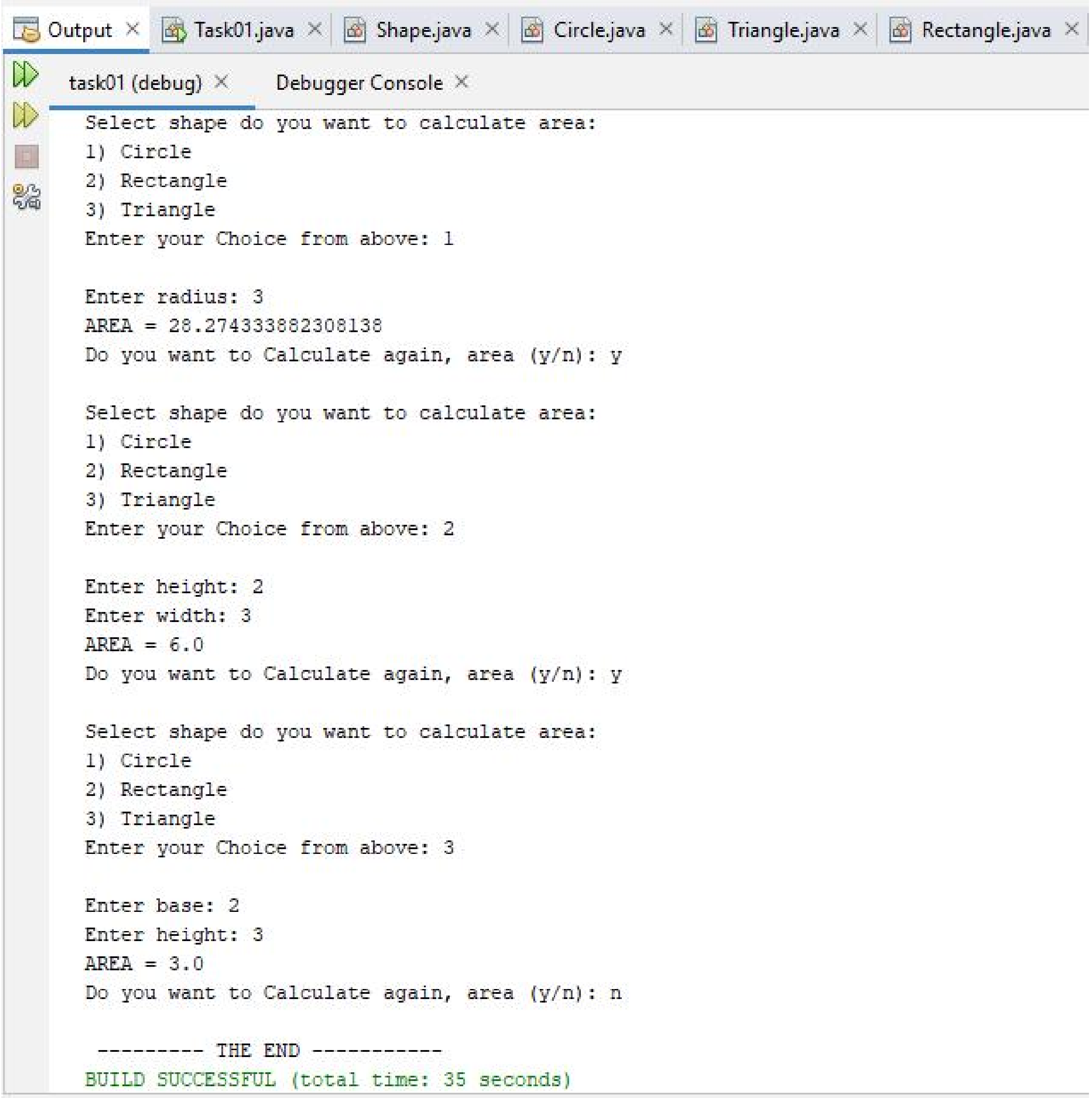
```
public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        char resp;

        Shape s = new Shape();

        do {

            System.out.println("\nSelect shape do you want to calculate area: ");

            System.out.println("1) Circle\n2) Rectangle \n3) Triangle");


            System.out.print("Enter your Choice from above: ");

            int res = input.nextInt();
```

```
System.out.println("");

switch (res) {

    case 1:             //circle

        Circle c = new Circle();

        System.out.print("Enter radius: ");

        double r1 = input.nextDouble();

        c.setRadius(r1);

        c.onAreaChange();

        break;

    case 2:             //rectangle

        Rectangle r = new Rectangle();

        System.out.print("Enter height: ");

        double h1 = input.nextDouble();

        System.out.print("Enter width: ");

        double w = input.nextDouble();

        r.setWidth(w);

        r.setheight(h1);

        r.onAreaChange();

        break;

    case 3:             //triangle

        Triangle t = new Triangle();


        System.out.print("Enter base: ");

        double b = input.nextDouble();

        System.out.print("Enter height: ");

        double h = input.nextDouble();

        t.setBase(b);

        t.setheight(h);
```

```
                  t.onAreaChange();
                  break;
              default:
                  System.out.println("Invalid entry -->Select (1)0r(2)0r(3) ");
                  break;
          }
          System.out.print("Do you want to Calculate again, area (y/n): ");
          resp = input.next().charAt(0);

      } while (resp == 'y' || resp == 'Y');
      System.out.println("\n --------- THE END -----------");
}
class Shape {
    protected double area;
    public Shape(){}
    public double getArea(){
        return area;
    }
    public void onAreaChange(){
        System.out.println("AREA = "+area);
    }
}
class Circle extends Shape {
  private double radius;
  Circle(){
      this.radius=0;
  }
  public void setRadius(double r){
      radius=r;
  }
  public double getArea(){
      area= Math.PI*radius*radius;
      return area;
  }
   public void onAreaChange(){
     getArea();
     super.onAreaChange();
  }
}
class Triangle extends Shape{
   private double base;
   private double height;
   Triangle(){
      this.base=0;
      this.height=0;
  }
  public void setBase(double b){
      this.base=b;
  }
  public void setheight(double h){
      this.height=h;
  }
  public double getArea(){
      area= (base*height)/2;
      return area;
  }
```

```
     public void onAreaChange(){
       getArea();
       super.onAreaChange();
    }
}
class Rectangle extends Shape{
      private double width;
     private double height;
     Rectangle(){
        this.width=0;
        this.height=0;
    }
    public void setWidth(double w){
        this.width=w;
    }
    public void setheight(double h){
        this.height=h;
    }
    public double getArea(){
        area= height*width;
        return area;
    }
     public void onAreaChange(){
       getArea();
       super.onAreaChange();
    }
}
```

Output:



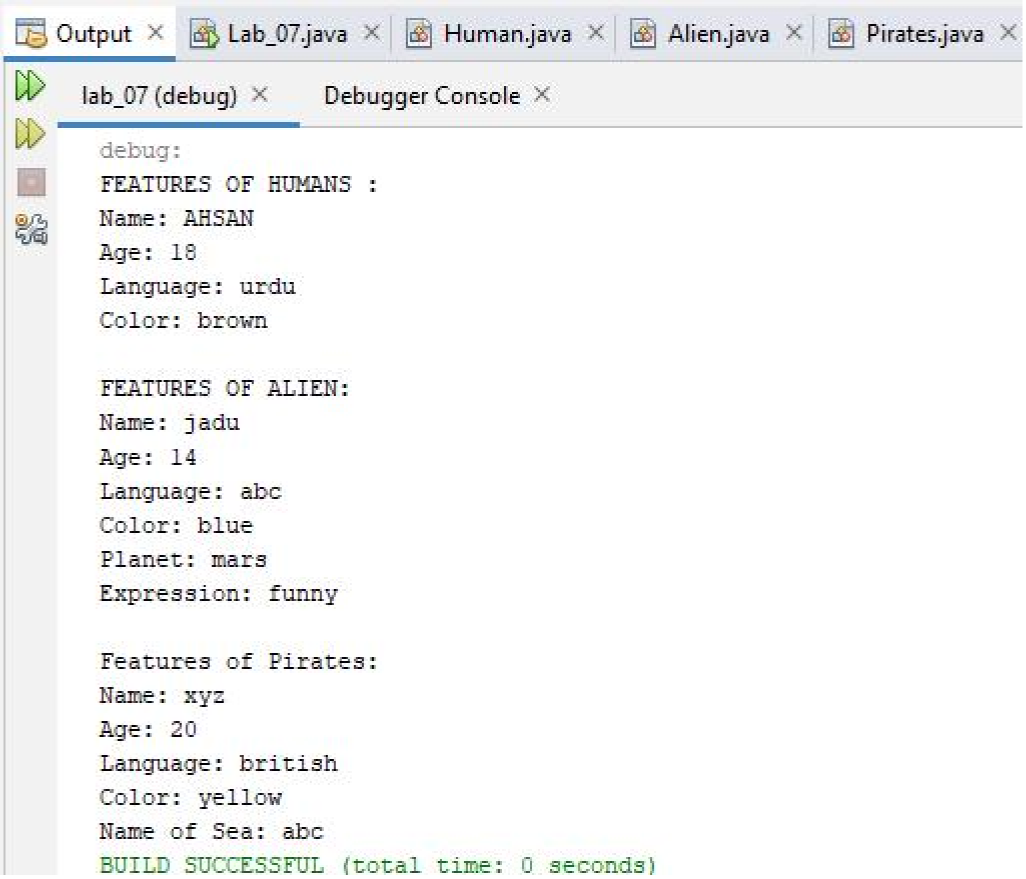
```
Select shape do you want to calculate area:
1) Circle
2) Rectangle
3) Triangle
Enter your Choice from above: 1

Enter radius: 3
AREA = 28.274333882308138
Do you want to Calculate again, area (y/n): y

Select shape do you want to calculate area:
1) Circle
2) Rectangle
3) Triangle
Enter your Choice from above: 2

Enter height: 2
Enter width: 3
AREA = 6.0
Do you want to Calculate again, area (y/n): y

Select shape do you want to calculate area:
1) Circle
2) Rectangle
3) Triangle
Enter your Choice from above: 3

Enter base: 2
Enter height: 3
AREA = 3.0
Do you want to Calculate again, area (y/n): n

 --------- THE END -----------
BUILD SUCCESSFUL (total time: 35 seconds)
```

Task # 02: Write a program that inherits a class named Alien and Pirates from a parent class Human. The human class has its own features like, Human can sleep, walk, talk etc. the Alien and Pirates class inheriting these functionalities as well as they have their characteristics, thus explaining the concepts of inheritance.

Solution:

```
public static void main(String[] args) {
        Human h = new Human("AHSAN", 18, "urdu", "brown");
        h.features();
        Alien a = new Alien("mars", "funny");
        a.name = "jadu";
        a.age = 14;
        a.language = "abc";
        a.skin_color = "blue";
        a.alienFeatures();
        Pirates p = new Pirates("abc");
        p.name = "xyz";
        p.age = 20;
        p.language = "british";
        p.skin_color = "yellow";
        p.piratesFeatures();
    }
class Human {
      protected String name;
    protected int age;
    protected String language;
    protected String skin_color;

    Human() {
        this.name = "";
        this.age = 0;
        this.language = "";
        this.skin_color = "";
    }
    Human(String n, int a, String l, String sc) {
            this.name = n;
            this.age = a;
            this.language = l;
            this.skin_color = sc;
    }
    public void features() {
            System.out.println("FEATURES OF HUMANS : ");
            System.out.println("Name: " + name);
            System.out.println("Age: " + age);
            System.out.println("Language: " + language);
            System.out.println("Color: " + skin_color);
}
class Alien extends Human {
          private String planet;
        private String expressions;
        Alien(){
                this.planet="";
                this.expressions="";
        }
        Alien(String p,String e){
                this.planet=p;
                this.expressions=e;
        }
        public void alienFeatures(){
                System.out.println("\nFEATURES OF ALIEN:");
  System.out.println("Name: " + name);
  System.out.println("Age: " + age);
  System.out.println("Language: " + language);
  System.out.println("Color: " + skin_color);
  System.out.println("Planet: " + planet);
  System.out.println("Expression: " + expressions);
      }
class Pirates extends Human {
        private String sea_name;
      public Pirates() {
  this.sea_name = "";
  }
  public Pirates(String sn) {
  this.sea_name = sn;
  }
  public void piratesFeatures() {
  System.out.println("\nFeatures of Pirates:");
  System.out.println("Name: " + name);
  System.out.println("Age: " + age);
  System.out.println("Language: " + language);
  System.out.println("Color: " + skin_color);
  System.out.println("Name of Sea: " + sea_name);
  }
```

Output:



```
debug:
FEATURES OF HUMANS :
Name: AHSAN
Age: 18
Language: urdu
Color: brown

FEATURES OF ALIEN:
Name: jadu
Age: 14
Language: abc
Color: blue
Planet: mars
Expression: funny

Features of Pirates:
Name: xyz
Age: 20
Language: british
Color: yellow
Name of Sea: abc
BUILD SUCCESSFUL (total time: 0 seconds)
```

Task # 03: Write a program that inherits a class named Produce, Cosmetics, Pharmacy, electronic Item and Cloth from a parent class Item. The Item class has its own features like, name and price etc. the Child classes inheriting these functionalities as well as they have their characteristics, thus explaining the concepts of inheritance. Chile classes like Produce, can have their own child classes i.e., Frozen and Fresh.

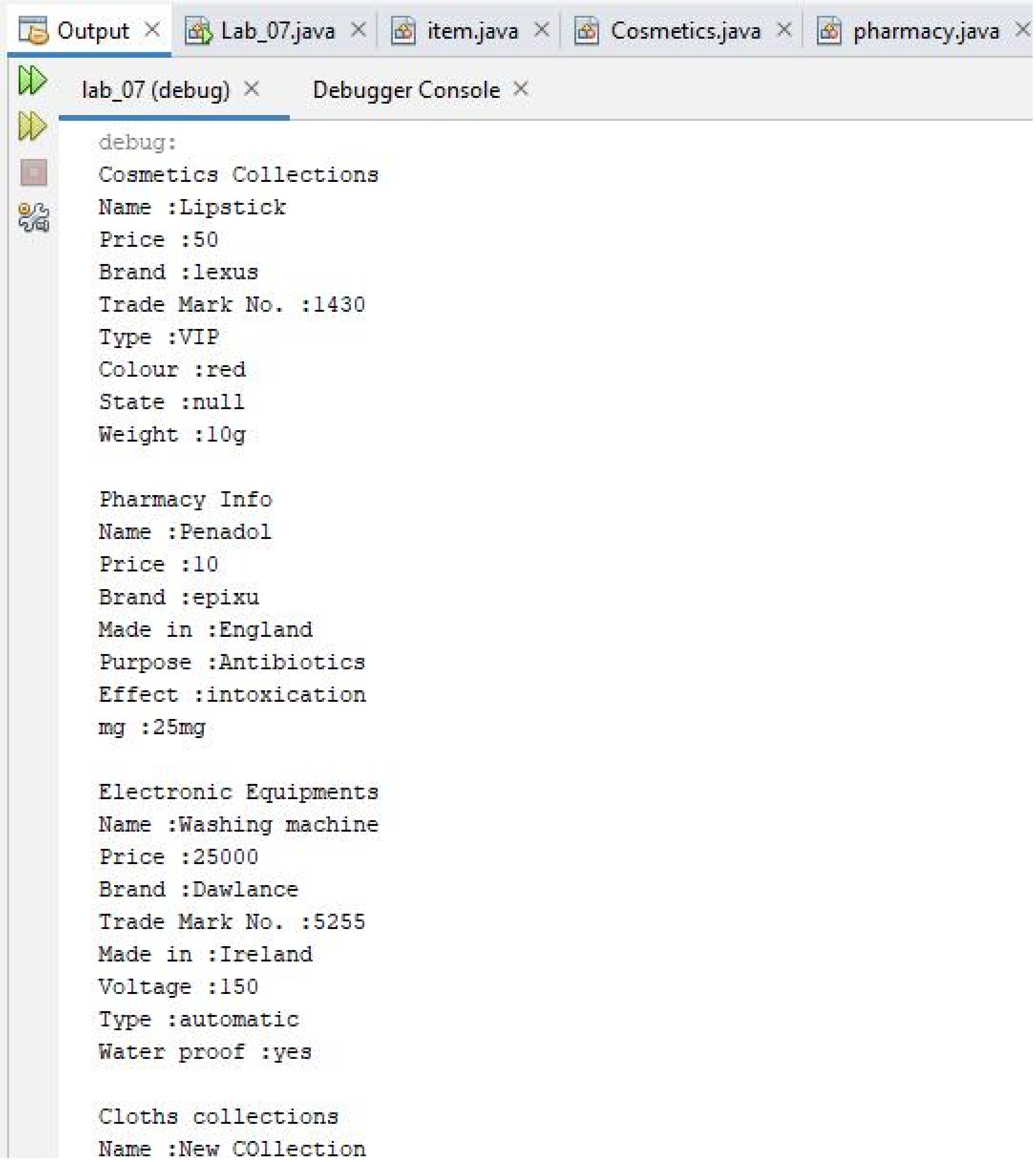
Solution:

```
 public static void main(String[] args) {

  System.out.println("Cosmetics Collections");

Cosmetics c1 = new Cosmetics("VIP", "red", "Semi_liquid", "10g");

c1.name="Lipstick";

c1.brand="lexus";

c1.made_in="Pakistan";
```

```
c1.TM_number=1430;

c1.price=50;

c1.display();

System.out.println("");

System.out.println("Pharmacy Info");

pharmacy p1=new pharmacy("Antibiotics", "25mg", "intoxication");

p1.name="Penadol";

p1.brand="epixu";

p1.made_in="England";

p1.price=10;

p1.display();

System.out.println("");

System.out.println("Electronic Equipments");

Electronics e1 = new Electronics("automatic", "yes", 150);

e1.name="Washing machine";

e1.price=25000;

e1.brand="Dawlance";

e1.made_in="Ireland";

e1.TM_number=5255;

e1.display();

System.out.println("");

System.out.println("Cloths collections");

clothes C2 = new clothes("Lawn", "Summer","upto you");

C2.name="New COllection";

C2.price=2500;

C2.brand="Ethnic";

C2.TM_number=1855;

C2.made_in="Pakistan";
```

```
 C2.display();
  }
  class Cosmetics extends item{
      private String type;
   private String colour;
   private String state;
   private String weight;
   Cosmetics() {}
   public Cosmetics(String type,String col,String State,String W)
   {
   this.type=type;
   this.colour=col;
   this.state=state;
   this.weight=W;
   }
   public void display()
   {
   System.out.println("Name :"+name);
   System.out.println("Price :"+price);
   System.out.println("Brand :"+brand);
   System.out.println("Trade Mark No. :"+TM_number);
   System.out.println("Type :"+type);
   System.out.println("Colour :"+colour);
   System.out.println("State :"+state);
   System.out.println("Weight :"+weight);
  }}
class pharmacy extends item {
    private String purpose;
 private String mg;

 private String effect;

 pharmacy(){}

 public pharmacy(String purpose,String mg,String effect)

 {

 this.purpose=purpose;

 this.effect=effect;

 this.mg=mg;

 }

 public void display()

 {

 System.out.println("Name :"+name);

 System.out.println("Price :"+price);

 System.out.println("Brand :"+brand);

 System.out.println("Made in :"+made_in);

 System.out.println("Purpose :"+purpose);

 System.out.println("Effect :"+effect);

 System.out.println("mg :"+mg);

 }

}

class Electronics extends item {
     private int volts;
 private String automatic_or_manual;
 private String water_res;
 public Electronics(){

 }
 public Electronics(String type,String waterproof,int volts)
 {
 this.volts=volts;
 this.water_res=waterproof;
 this.automatic_or_manual=type;
 }
 public void display()
 {
 System.out.println("Name :"+name);
 System.out.println("Price :"+price);
 System.out.println("Brand :"+brand);
 System.out.println("Trade Mark No. :"+TM_number);
 System.out.println("Made in :"+made_in);
 System.out.println("Voltage :"+volts);
 System.out.println("Type :"+automatic_or_manual);
 System.out.println("Water proof :"+water_res);
 }}
class clothes extends item{
        private String type;
        private String season;
        private String choice;
        public clothes(){}
        public clothes(String t,String s,String c){
                this.type=t;
              this.season=s;
              this.choice=c;
        }
   public void display()
 {
 System.out.println("Name :"+name);
 System.out.println("Price :"+price);
 System.out.println("Brand :"+brand);
 System.out.println("Trade Mark No. :"+TM_number);
 System.out.println("Made in :"+made_in);
 System.out.println("Type :"+type);
 System.out.println("Season :"+season);
 System.out.println("choice :"+choice);
 }

}
```

Output:



```
debug:
Cosmetics Collections
Name :Lipstick
Price :50
Brand :lexus
Trade Mark No. :1430
Type :VIP
Colour :red
State :null
Weight :10g

Pharmacy Info
Name :Penadol
Price :10
Brand :epixu
Made in :England
Purpose :Antibiotics
Effect :intoxication
mg :25mg

Electronic Equipments
Name :Washing machine
Price :25000
Brand :Dawlance
Trade Mark No. :5255
Made in :Ireland
Voltage :150
Type :automatic
Water proof :yes

Cloths collections
Name :New COllection
```

Task # 04: Write a program that inherits a class named Pakistani, BBQ, Chines, Fast Food and Beverages etc. from a parent class Cuisines. The Cuisines class has its own features like, name, quantity and price etc. the Child classes inheriting these functionalities as well as they have their characteristics, thus explaining the concepts of inheritance. Child classes can have their own child classes.

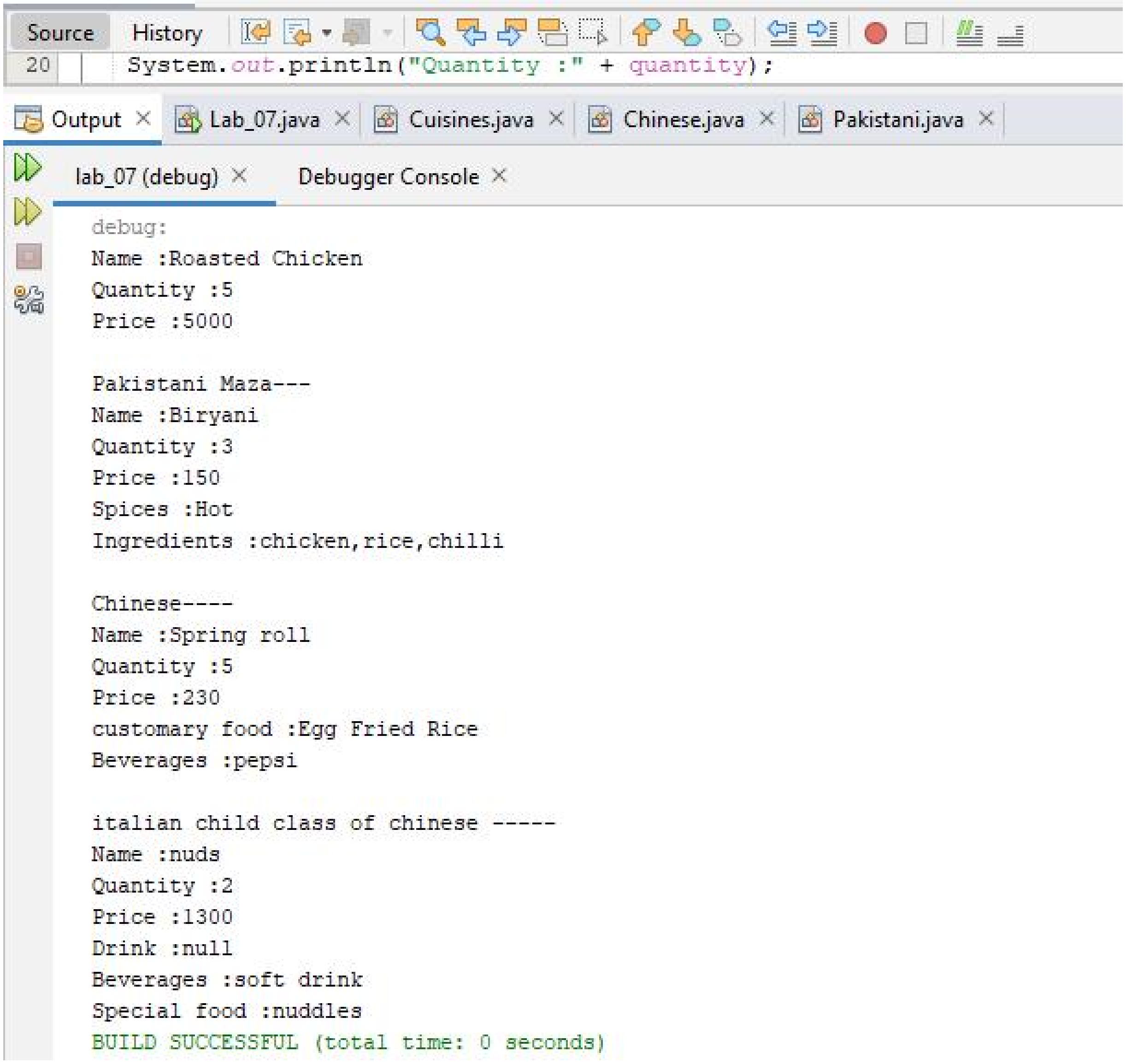
Solution:

```
public static void main(String[] args) {
Cuisines cu1 = new Cuisines("Roasted Chicken", 5, 5000);
  cu1.display();
  System.out.println("");
  System.out.println("Pakistani Maza---");
  Pakistani p1 = new Pakistani("Hot", "chicken,rice,chilli");
  p1.name = "Biryani";
  p1.price = 150;
  p1.quantity = 3;
  p1.display();
  System.out.println("");
  System.out.println("Chinese----");
  Chinese c1 = new Chinese("Egg Fried Rice", "pepsi");
  c1.name = "Spring roll";
  c1.price = 230;
  c1.quantity = 5;
  c1.display();
                italian i1=new italian("nuddles","scray");
                System.out.println("\nitalian child class of chinese -----");
i1.name="nuds";
i1.quantity=2;
i1.price=1300;
i1.beverages="soft drink";
i1.display();
    }
    public class Cuisines {
          protected String name;
      protected int quantity;
      protected int price;
      public Cuisines() {
      }
      public Cuisines(String name, int quant, int pric) {
      this.name = name;
      this.price = pric;
      this.quantity = quant;
      }
      public void display() {
      System.out.println("Name :" + name);
      System.out.println("Quantity :" + quantity);
      System.out.println("Price :" + price);
      }
        class Chinese extends Cuisines {
           String customary_food;
      String beverages;
      Chinese() {
      }
      Chinese(String cutom, String bever) {
      this.beverages = bever;
      this.customary_food = cutom;
      }
      public void display() {
      System.out.println("Name :" + name);
      System.out.println("Quantity :" + quantity);
      System.out.println("Price :" + price);
      System.out.println("customary food :" + customary_food);
      System.out.println("Beverages :" + beverages);
```

```
  }}
class Pakistani extends Cuisines {
        String spices;
  String ingridients;
  public Pakistani() {
  }
  public Pakistani(String spicies, String ingred) {
  this.spices = spicies;
  this.ingridients = ingred;
  }
  public void display() {
  System.out.println("Name :" + name);
  System.out.println("Quantity :" + quantity);
  System.out.println("Price :" + price);
  System.out.println("Spices :" + spices);
  System.out.println("Ingredients :" + ingridients);
  }
}
class italian extends Chinese{
        private String specialfood;
        private String drink;
        italian(String s,String d){
                this.specialfood=s;
                this.drink=d;
        }
        public void display() {
  System.out.println("Name :" + name);
  System.out.println("Quantity :" + quantity);
```

```
   System.out.println("Price :" + price);

   System.out.println("Drink :" + customary_food);

   System.out.println("Beverages :" + beverages);

 System.out.println("Special food :" + specialfood);

        }
```

Output:



```
System.out.println("Quantity :" + quantity);
```

```
debug:
Name :Roasted Chicken
Quantity :5
Price :5000

Pakistani Maza---
Name :Biryani
Quantity :3
Price :150
Spices :Hot
Ingredients :chicken,rice,chilli

Chinese----
Name :Spring roll
Quantity :5
Price :230
customary food :Egg Fried Rice
Beverages :pepsi

italian child class of chinese -----
Name :nuds
Quantity :2
Price :1300
Drink :null
Beverages :soft drink
Special food :nuddles
BUILD SUCCESSFUL (total time: 0 seconds)
```

Task # 05: Write code according to given guide.You must draw a class diagram first to start

writing your code.

Consider a superclass Items which models customer's purchases. This class has:

- Two private instance variables name (String) and unitPrice (double).
- One constructor to initialize the instance variables.
- A default constructor to initialize name to "no item", and unitPrice to 0. use this()
- A method getPrice that returns the unitPrice.
- Accessor and mutator methods.
- A toString method to return the name of the item followed by @ symbol, then the

unitPrice.

Consider two subclasses WeighedItem and CountedItem. WeighedItem has an additional

instance variable weight (double) in Kg while CountedItem has an additional variable quantity

(int) both private.

Write an appropriate constructor for each of the classes making use of the constructor of the

superclass in defining those of the subclasses.

Override getPrice method that returns the price of the purchasedItem based on its unit price and

weight (WeighedItem), or quantity (CountedItem). Make use of getPrice of the superclass

Override also toString method for each class making use of the toString method of the

superclass in defining those of the subclasses.

toString should return something that can be printed on the receipt.

For example

Banana @ 3.00 1.37 Kg 4.11 PKR (in case of WeighedItem class)

Pens @ 4.5 10 units 45 PKR (in case of CountedItem class)

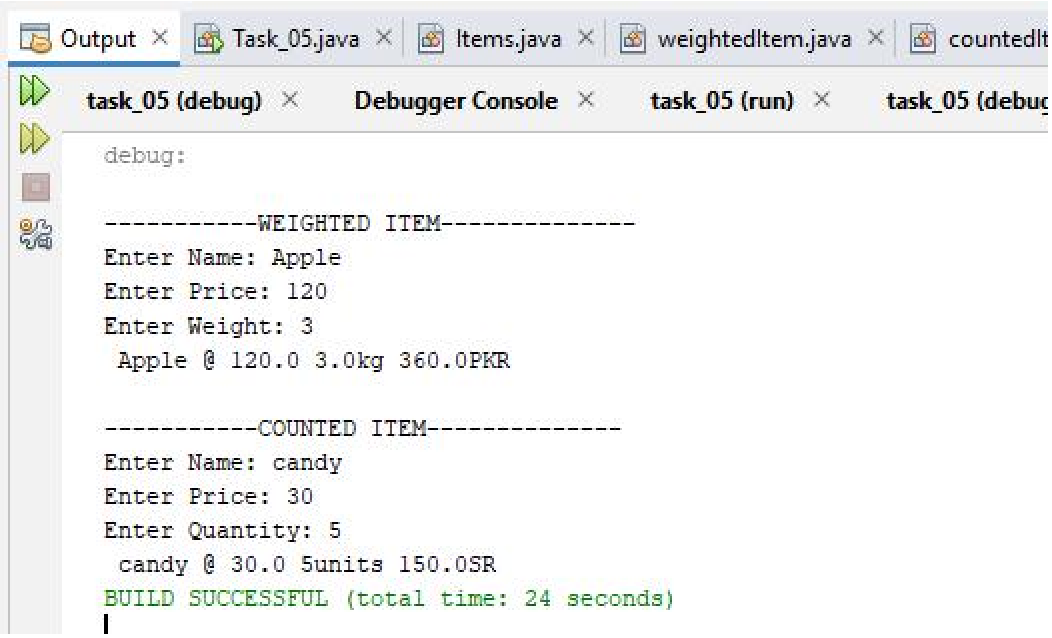
Solution:

```
  public static void main(String[] args) {
              Scanner scan = new Scanner(System.in);
              System.out.println("\n-----------WEIGHTED ITEM--------------");
              System.out.print("Enter Name: ");
              String n = scan.nextLine();
              weightedItem w = new weightedItem();
              w.setName(n);

              System.out.print("Enter Price: ");
              double p = scan.nextDouble();
              w.setPrice(p);
              System.out.print("Enter Weight: ");
              double w1 = scan.nextDouble();
              w.setWeight(w1);
              System.out.println(" " + w.toString());
              countedItem c = new countedItem();

System.out.println("\n-----------COUNTED ITEM--------------");
              System.out.print("Enter Name: ");
              String n1 = scan.next();
              c.setName(n1);
              System.out.print("Enter Price: ");
              double p1 = scan.nextDouble();
              c.setPrice(p1);
              System.out.print("Enter Quantity: ");
              int q = scan.nextInt();
              c.setQuantity(q);
              System.out.println(" " + c.toString());

   }
   public class Items {

          private String name;
          private double unitprice;

          Items() {
                this.name = "no item";
```

```
        this.unitprice = 0;
    }

    Items(String n, double p) {
        this.name = n;
        this.unitprice = p;
    }

    public double getPrice() {
        return unitprice;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setPrice(double p) {
        this.unitprice = p;
    }

    public String toString() {
        return name + " @ " + unitprice;
    }
}
public class weightedItem extends Items {
    private double weight;
    public weightedItem() {
    this.weight=0;
    }
    public weightedItem(double weight) {
        this.weight = weight;
    }
    public double getWeight() {
        return weight;
    }

    public void setWeight(double weight) {
        this.weight = weight;
    }

    public double getPrice() {
        return weight * super.getPrice();
    }

    public String toString() {
        return super.toString() + " " + weight + "kg " + getPrice() + "PKR";
    }
```

```
}
public class countedItem extends Items {

        private int quantity;

        public countedItem() {
        }
        public countedItem(int quantity) {
                this.quantity = quantity;
        }
        public int getQuantity() {
                return quantity;
        }
        public void setQuantity(int quantity) {
                this.quantity = quantity;
        }
        public double getPrice() {
                return quantity * super.getPrice();
        }
        public String toString() {
                return super.toString() + " " + quantity + "units " + getPrice() + "SR";
        }
}
```

Output:



```
debug:

-----------WEIGHTED ITEM--------------
Enter Name: Apple
Enter Price: 120
Enter Weight: 3
 Apple @ 120.0 3.0kg 360.0PKR

-----------COUNTED ITEM--------------
Enter Name: candy
Enter Price: 30
Enter Quantity: 5
 candy @ 30.0 5units 150.0SR
BUILD SUCCESSFUL (total time: 24 seconds)
```