The code is :

```
// For class FrontPage, you need to implement Four types
// of member function
class FrontPage implements Cloneable {
  private int numOfDishes=80; // Number of dishes listed on the front page
  // Default value is 80
  // Manager function
  public FrontPage(int numOfDishes1) {
  numOfDishes=numOfDishes1;
  }
  public Object clone() {
  try
  {
  return super.clone();
  }
  catch (CloneNotSupportedException e)
  {
  return null;
  }
  }
  // Access function
  // get
  public int getNumOfDishes() {
  return numOfDishes;

  }
  // set
  public void setNumOfDishes(int numOfDishes1) {
  numOfDishes = numOfDishes1;

  }
  // predicate
  public boolean isLargeRestaurent() {
  return numOfDishes > 200;
  }
  // Implementor
  public void increaseNumOfDishes(int num) {
  numOfDishes += num;
  }
```

```
  public String toString() {
  return "Number of dishes: " + numOfDishes + "\n";
  }
  public boolean equals(FrontPage that) {
       if (!(that instanceof FrontPage)) return false;

       FrontPage tstA;
       tstA = (FrontPage) that;
       return (numOfDishes == that.numOfDishes);
  }




}
// For class BackPage, you need to implement Four types
// of member function
class BackPage implements Cloneable {
  private String orderNum; // Phone number to order this kind of food.
  // E.g., 408-234-5678, 408-867-5234
  // Manager function
  public BackPage(String orderNum1) {
  orderNum = orderNum1;
  }
  public Object clone() {
  try
  {
  return super.clone();
  }
  catch (CloneNotSupportedException e)
  {
  // This shouldn't happen, since we are Cloneable
  return null;
  }
  }
  // Access function
  // get
  public String getOrderNum() {
  return orderNum;

  }
  // set
```

```
  public void setOrderNum(String orderNum1) {
  orderNum = orderNum1;

  }
  // predicate
  public boolean isUnique() {
  return orderNum.equals("408-111-1111");
  }
  // Implementor
  public void changeOrderNum(String orderNum1) {
  setOrderNum(orderNum1);

  }




  public String toString() {
  return "Phone number to order: " + orderNum + "\n";
  }
  public boolean equals(BackPage that) {
       if (!(that instanceof BackPage)) return false;

       BackPage tstA;
       tstA = (BackPage) that;
       return (orderNum == that.orderNum);
  }




}
// For class Menu, you need to implement Four types of member function
class Menu implements Cloneable {
  private String title="Ammerican Food";
  // The title of the Menu
  // For example, "Mexican Food", "Ammerican Food",
// "Indian Food", etc.
// Default is "Ammerican Food".
  private FrontPage fp = new FrontPage(80);
  // Default value for fp's numOfDishes is 80
```

```
private BackPage bp = new BackPage("408-123-4567");
// Default value for bp's orderNum is "408-123-4567"
// Manager function
```

```
public Menu(String title1, FrontPage fp1, BackPage bp1) {
title = title1;
fp = (FrontPage)fp1.clone();
bp = (BackPage)bp1.clone();
}
public Object clone()
{
      try
       {
            Menu mobj = (Menu)super.clone();
            mobj.fp = (FrontPage)fp.clone();
            mobj.bp = (BackPage)bp.clone();
            return mobj;
       }
       catch (CloneNotSupportedException e)
       {
            return null;
       }
}
```

```
//Access functions
public FrontPage getFp() {
return fp;
}
public void setFp(int numOfDishes1) {
fp.setNumOfDishes(numOfDishes1);
}
public BackPage getBp() {
return bp;
}
public void setBp(String orderNum1) {
bp.setOrderNum(orderNum1);
```

```
}
public String getTitle() {
return title;
}
public void setTitle(String title1) {
title = title1;
}
public boolean isAmericanFood() {
return title.equals("Ammerican Food");
}
// Implementor functions
public void changeToMexicanFood() {
setTitle("Mexican Food");
}
```

```
public String toString() {
return ("Title is: " + title + "\n" + fp.toString() + bp.toString());
}
public boolean equals(Menu that) {
    if (!(that instanceof Menu)) return false;

    Menu tstA;

    tstA = (Menu) that;

    return (title.equals(tstA.title) && fp.equals(tstA.fp) && bp.equals(tstA.bp));
}
```

```
public Menu getThisMenu() {
return this;
}
}
```

```
// For class FancyMenu, you need to implement Four types of member functions
class FancyMenu extends Menu implements Cloneable {
  private String color="white";
  // The color of the FancyMenu
// For example, "red", "green", etc.
  // Manager function



  public FancyMenu(String color1, String title1, FrontPage fp1, BackPage bp1)
  {
  super(title1, fp1, bp1);
  color = color1;
  }



  //Access functions
  public String getColor() {
  return color;
  }
  public void setColor(String color1) {
  color = color1;
  }
  public boolean isRed() {
  return color.equals("red");
  }
  // Implementor functions
  public void changeColor(String color1) {
  setColor(color1);
  }




  public String toString() {
  return super.toString() + "color: " + color;
  }
  public boolean equals(FancyMenu that) {
        if (!(that instanceof FancyMenu)) return false;

        FancyMenu tstE;
        tstE = (FancyMenu) that;
```

```
        return (color == tstE.color && (super.equals((Menu)tstE)));
    }



}
// After the classes are created, create two Menu objects
// in the main() function:
// Menu 1: Tille "Mexican Food"
// Number of dishes listed on the fron numOfDishes: 34
// Phone number to order: 408-234-5678
// Menu 2: Tille "American Food"
// Number of dishes listed on the fron numOfDishes: 15
// Phone number to order: 408-524-6789
//
// Will display:
// Title: Mexican Food
// Number of dishes: 34
// Phone number to order: 408-234-5678-4567
//
// ============================
// Title: American Food
// Number of dishes: 15
// Phone number to order: 408-524-6789
//
// ============================
// Cloning menu3 from menu2 is successful
// ============================
// menu3 is not equal to menu1
// ============================
// fm1 is not equal to fm2
// ============================
// Cloning fm3 from fm2 is successful
// ============================
// Title: Mexican Food
// Number of dishes: 34
// Phone number to order: 408-234-5678-4567
// Color: red
// ============================
// Title: American Food
// Number of dishes: 15
// Phone number to order: 408-524-6789
// Color: blue
```

```
// ============================
public class MenuDemo {
  public static void main(String args[]) {
  FrontPage fp1 = new FrontPage(34);
  BackPage bp1 = new BackPage("408-234-5678-4567");
  Menu menu1= new Menu("Mexican Food", fp1, bp1);
  System.out.println(menu1);
  System.out.println("============================");
  FrontPage fp2 = new FrontPage(15);
  BackPage bp2 = new BackPage("408-524-6789");
  Menu menu2= new Menu("American Food", fp2, bp2);
  System.out.println(menu2);
  System.out.println("============================");
  Menu menu3 = (Menu)menu2.clone();
  if (menu3.equals(menu2))
  System.out.println("Cloning menu3 from menu2 is successful");
  else
  System.out.println("Cloning menu3 from menu2 is not successful");
  System.out.println("============================");
  if (menu3.equals(menu1))
  System.out.println("menu3 is equal to menu1");
  else
  System.out.println("menu3 is not equal to menu1");
  System.out.println("============================");
  FancyMenu fm1 = new FancyMenu("red", "Mexican Food", fp1, bp1);
  FancyMenu fm2 = new FancyMenu("blue", "American Food", fp2, bp2);
  if (fm1.equals(fm2))
  System.out.println("fm1 is equal to fm2");
  else
  System.out.println("fm1 is not equal to fm2");
  System.out.println("============================");
  FancyMenu fm3 = (FancyMenu)fm2.clone();
  if (fm3.equals(fm2))
  System.out.println("Cloning fm3 from fm2 is successful");
  else
  System.out.println("Cloning fm3 from fm2 is not successful");
  System.out.println("============================");
  System.out.println(fm1);
  System.out.println("============================");
  System.out.println(fm2);
  System.out.println("============================");
  }
}
```

```
339   else
340   System.out.println("Cloning fm3 from fm2 is not successful");
341   System.out.println("==========================");
342   System.out.println(fm1);
343   System.out.println("==========================");
344   System.out.println(fm2);
345   System.out.println("==========================");
346   }
347 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4 | Interactive | Stdin Inputs

CommandLine Arguments

Execute

Result

CPU Time: 0.34 sec(s), Memory: 37860 kilobyte(s) — compiled and executed in 0.983 sec(s)

```
Title is: Mexican Food
Number of dishes: 34
Phone number to order: 408-234-5678-4567

==========================
Title is: American Food
Number of dishes: 15
Phone number to order: 408-524-6789

==========================
Cloning menu3 from menu2 is successful
==========================
menu3 is not equal to menu1
==========================
fm1 is not equal to fm2
==========================
Cloning fm3 from fm2 is successful
==========================
Title is: Mexican Food
Number of dishes: 34
Phone number to order: 408-234-5678-4567
color: red
==========================
Title is: American Food
Number of dishes: 15
Phone number to order: 408-524-6789
color: blue
==========================
```