

SCSJ2154
REVISION EXERCISES

1. Given the following Program B1;

1	//Program B1
2	class ClassA{
3	public ClassA() {}
4	public void method1()
5	{ System.out.println("UTM"); }
6	public void method1(String a)
7	{ System.out.println("UTM" +a); }
8	public void method1(int a)
9	{ System.out.println("UTM" +a); }
10	}
11	class ClassB extends ClassA{
12	public ClassB() {}
13	public void method1()
14	{ System.out.println("FC UTM"); }
15	public void method1(String a)
16	{ System.out.println("FC UTM" +a); }
17	public void method2(String a, int b)
18	{ System.out.println("Studied at "+a+" in "+b); }
19	}
20	class ClassC extends ClassB{
21	public ClassC() {}
22	public void method1()
23	{ System.out.println("SE@FC UTM"); }
24	public void method1(int a)
25	{ System.out.println(" SE@FC UTM" +a); }
26	}
27	
28	public class TestFC {
29	public static void main(String []args)
30	{
31	_____
32	_____
33	}
34	}

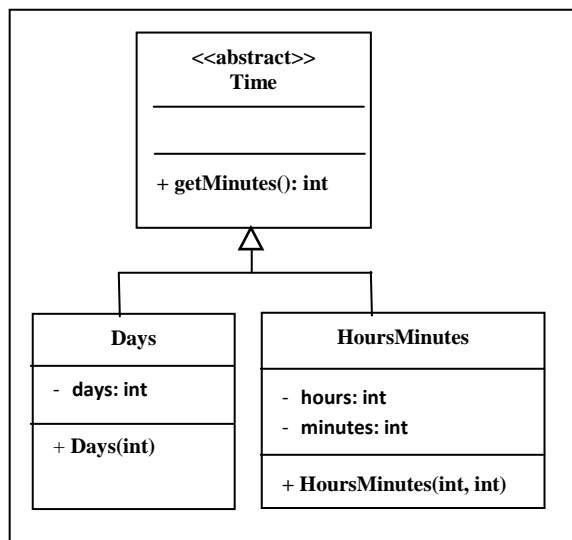
If the following statements are inserted at line 31 and 32, determine whether the program is correct or has an error during compilation. If the program is correct, state the output. If the program has an error, give the reason. Write your answer as in Table 1.

- a) ClassA ob = new ClassC();
 ob.method1(2017);
- b) ClassA ob = new ClassC();
 ob.method1("JB");
- c) ClassA ob = new ClassB();
 ob.method2("FC UTM",2017);
- d) ClassC ob = new ClassB();
 ob.method1();
- e) ClassC ob = new ClassC();
 ob.method2("FSKSM UTM",1997);

Table 1

Statement No.	Correct / Error	Output/Reason
a)		
b)		
c)		
d)		
e)		

2. Figure B1 shows relationship of the classes in Program B2. Write the missing Java statements in Program B2 as guided in the comment parts in order to implement the class hierarchy as in Figure B1.

**Figure B1**

```

1. // Program B2
2. _____(i)_____ // Declaration of abstract class Time
3. {
4. _____(ii)_____ // with an abstract method
5. } // getMinutes()
6.
7. _____(iii)_____ // Signature of class Days that
8. { // inherits class Time
9.     private int days;
10.     _____(iv)_____ // Parameterized constructor
11.     _____(v)_____ // of class Days
12.
13.     public int getMinutes() {
14.         return days * 24 * 60;
15.     }
16. }
17.
18. _____(vi)_____ // Signature of class HoursMinutes that
19. { // inherits class Time
20.     private int hours;
21.     private int minutes;
22.
23.     _____(viii)_____ // Parameterized constructor
24.     _____(ix)_____ // of class HoursMinutes
25.     _____(x)_____
26.
27.     public int getMinutes() {
28.         return hours * 60 + minutes;
29.     }
30. }
31.
32. public class Demo {
33.     public static void main(String args[]) {
34. // Create an object of class Time that refer to class Days
35. // named t1 with argument 2
36.     _____(xi)_____
37. // Create an object of class Time that refer to class HoursMinutes
38. // named t2 with arguments 4 and 10
39.     _____(xii)_____
40.     System.out.println(t1.getMinutes());
41.     System.out.println(t2.getMinutes());
42. }
43. }

```

Question 3

Given the UML class diagram in Figure B2, Program B3, and output in Figure B3, answer the following questions (a) to (c).

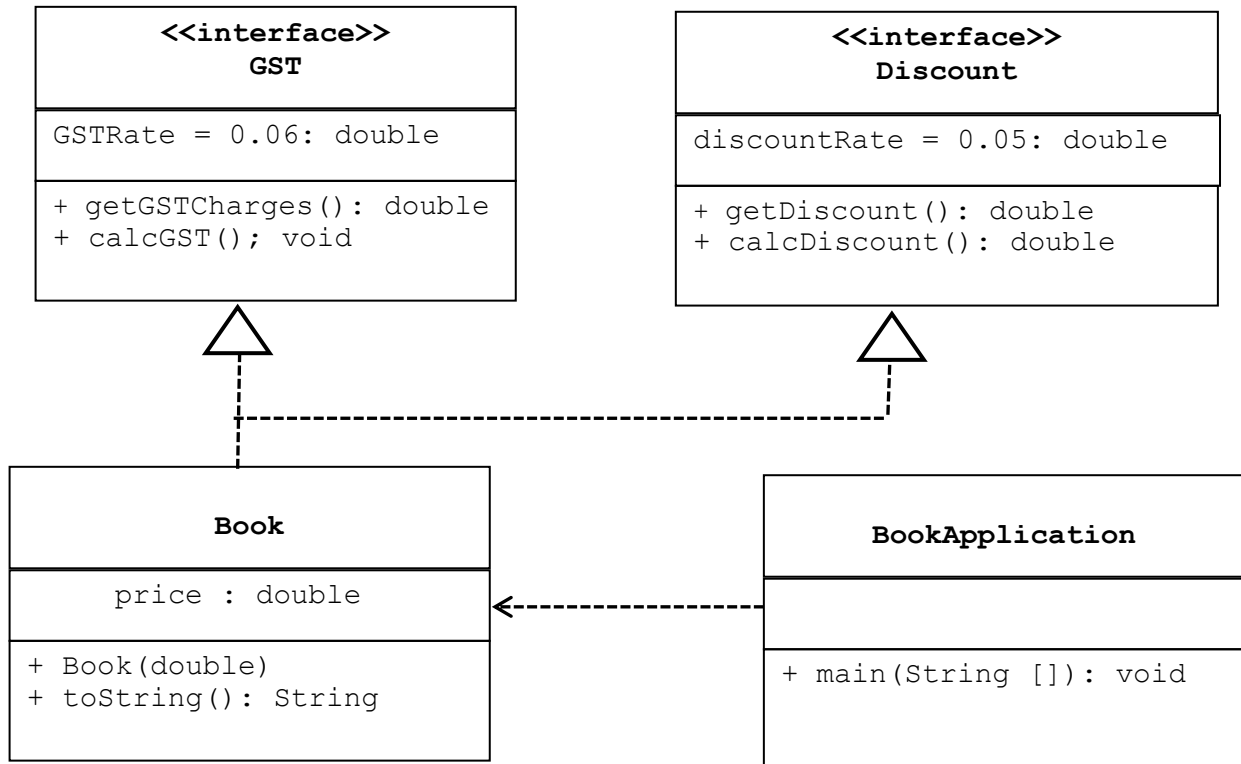


Figure B2: The UML class diagram

```
C:\Windows\system32\cmd...
Initial Price: 10.0
Price after 5% discount: 9.5
Price after discount and GST: 10.1
Press any key to continue . . .
```

Figure B3 : Output of Program B3

```

1 //Program B3
2
3 _____(a)_____{
4 _____
5 _____
6 _____
7 _____
8 }
9
10 _____ {
11 _____
12 _____
13 _____
14 _____
15 _____
16 }
17
18 _____(b)_____{
19 _____
20 _____
21 _____
22 }
23
24 public String toString(){
25     return "\nInitial Price: "+price+"\nPrice after 5%
26 discount: "+ (price-calcDiscount())+ "\nPrice after discount and
27 GST: "+(price-calcDiscount()+getGSTCharges());
28 }
29 public double getGSTCharges() {return price*RATE;}
30 public double calcGST() {return price+getGSTCharges();}
31 public double getDiscount() { return rate;}
32 public double calcDiscount() { return price*getDiscount();}
33 }
34
35 public class BookApplication {
36     public static void main(String[] args) {
37         _____(c)_____
38         _____(d)_____
39     }
40 }
41 }

```

- (i) Write Java code that defines GST (line 3-7) and Discount (line 11-15) interface classes.
- (ii) Write Java code that defines class Book (line 18-22) that implements the interfaces defined in (a).
- (iii) Write Java code to create a Book object with price is initialized with 10.
- (iv) Display the price of book after discounts and tax levied GST by invoking toString() method.

Question 4

Answer question (i) to (v) as in Program B4 below with suitable codes so that it can throw the exception. [5 marks]

```
//Program B4
public class FinalExamException {
    public static void main (String args[]) {
        int arr [] = {30,40};
        Scanner in = new Scanner (System.in);
        ____ (i) ____ {
            int b = in.nextInt();
            int x = arr[2]/ (b - arr[1]);
        }

        catch (____ (ii) _____ ex) {
            System.out.println("Exceed array size");
        }
        catch (____ (iii) _____ ex) {
            System.out.println("Denominator is zero");
        }

        catch (____ (iv) _____ ex) {
            System.out.println("Invalid data:" +e);
        }
        ____ (v) _____ {
            int y = arr[1]/ arr[0];
            System.out.println("y = " +y);
        }
    }
}
```

a. Given Program B5 below, answer the following question.

- (i) Explain why error will occur when Program B5 is compiled?
- (ii) Rewrite the program so that the program will compile and run properly.

```
//Program B5

class Test {
    public static void main(String[] args) {
        try {
            String s = "5.6";
            Integer.parseInt(s);

            int i = 0;
            int y = 2 / i;
        }
        catch (Exception ex) {
            System.out.println("NumberFormatException");
        }
        catch (RuntimeException ex) {
            System.out.println("RuntimeException");
        }
    }
}
```

- b. Complete class **Example2** by answering the question (i) to (x) based on the comments given in Program B4. (7 marks)

```
// Program B4
1 import java.util.Scanner;
2
3 public class Example2 {
4     public static void main(String args[]) {
5         try {
6             Scanner input = new Scanner(System.in);
7             String _____(i)_____; // declare a string variable
8             _____(ii)_____; // declare another related variable
9             System.out.println ("Enter an integer:");
10            inputData = input.nextLine();
11            num = Integer.parseInt(inputData);
12            System.out.println ("The square is: "+(num*num));
13            method();
14        }
15        _____(iii)_____ (_____(iv)_____ ex) {
16            System.out.println ("Wrong data type");
17        }
18        _____(v)_____ (_____(vi)_____ ex) {
19            System.out.println ("Array index out of bound");
20        }
21        _____(vii)_____ (Exception ex) {
22            System.out.println ("Input problem");
23        }
24    }
25
26    _____(viii)_____ void method() _____(ix)_____ Exception {
27        String languages[] = { "C++", "Java", "Perl"};
28
29        for (int i = 1; i <= 3; i++) {
30            System.out.println(languages[i]);
31        }
32    }
33 }
```

- (i) What is the expected output if the input entered is 4?

LONG PROGRAMMING

Given the following class diagram in Figure 6, sample output in Figure 7, and the description method of classes in Table 3, answer all questions in this part.

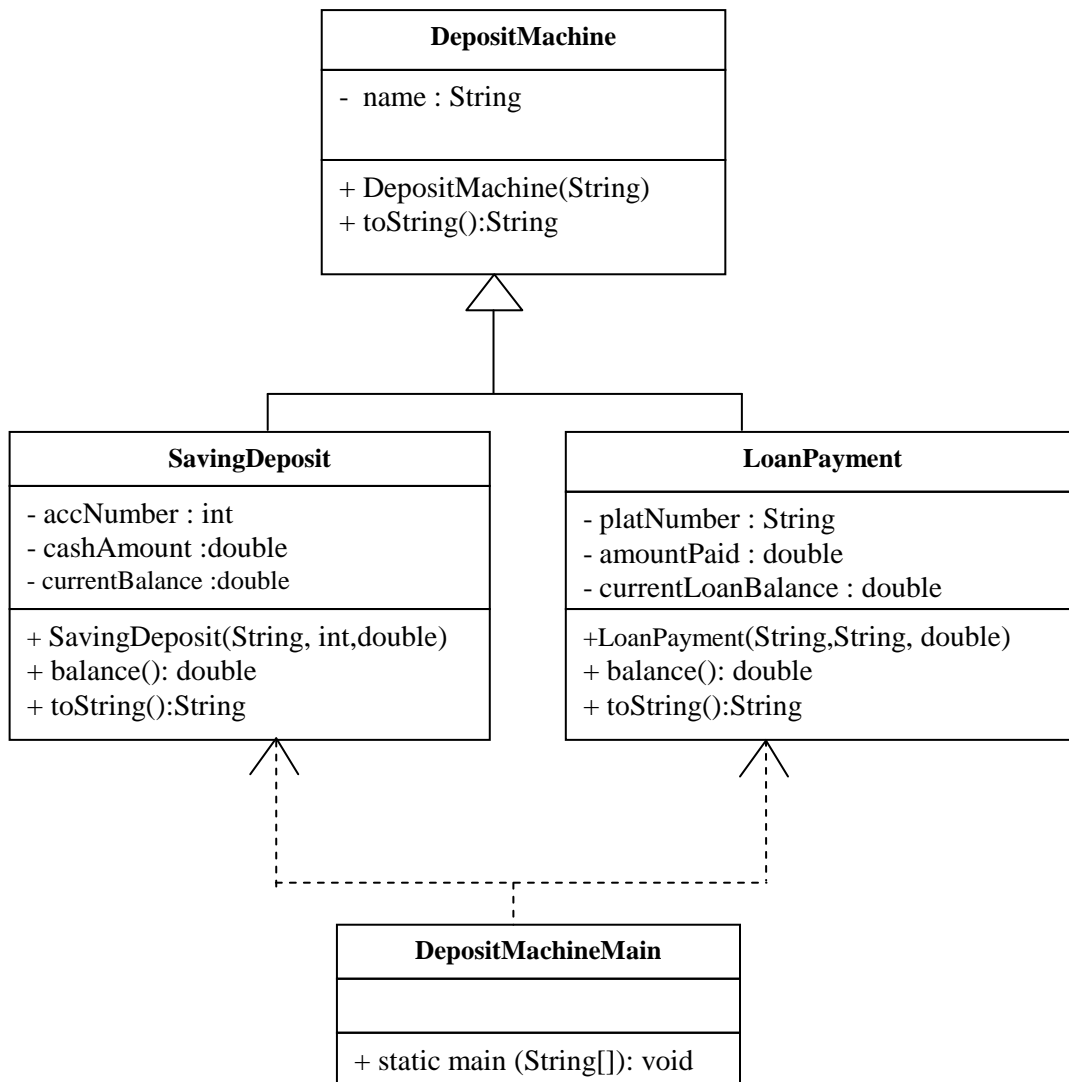


Figure 6

```

Name: Ali
Deposit: RM100.0 to Account: 777
Account Balance: RM5100.0
Name: Abu
Amount Paid: RM1000.0 for Vehicle: JKE222
Loan Balance: RM4000.0
    
```

Figure 7: Sample Output

Table 3

Method	Description
DepositMachine(String)	A superclass constructor that receives name of type String
toString()	A superclass method which concatenate label name and name, and return as a String
SavingDeposit(String, int, double)	A subclass constructor which receives depositor name , accNumber and cashAmount . The input name will pass to superclass constructor

balance()	A SavingDeposit subclass method which calculates (currentBalance + cashAmount) and return an account balance of type double
toString()	A SavingDeposit subclass method which concatenate name, saving amount, account number and balance and return as a single string
LoanPayment(String, String, double)	A subclass constructor that receives loan payer name, the vehicle plat number and amount paid
balance()	A LoanPayment subclass method which calculates (currentLoanBalance - amountPaid) and return a loan balance of type double
toString()	A LoanPayment subclass method which concatenate name, amount paid, plat number and balance and return as a single string
main(String[] args)	A main method that instantiate two objects of class SavingDeposit and LoanPayment . The method print out all the output as shown in the above sample output snapshot

Write a complete Java program that contains classes as below:

- i. The **DepositMachine** class serves as the parent of all classes and contains information that applies to all depositors. [5 Marks]
- ii. The **SavingDeposit** class represents a depositor that earns an additional cash money in his/her account. The **SavingDeposit** class contains of three instance variables i.e. **accNumber** that represents his/her account bank number, the amount of cash deposit which is **cashAmount** and **currentBalance** that represents the latest balance in his/her account before deposit transaction. The **balance** method returns the balance after deposit transaction (**currentBalance + cashAmount**) . [9 Marks]
- iii. The **LoanPayment** class represents a depositor that pays his/her bank loan. It contains an instance variable i.e. **platNumber** that represents the vehicle registration number, the amount paid which is **amountPaid**, **currentLoanBalance** which represents the loan balance before payment transaction. The **balance** method returns the balance after the transaction (**currentLoanBalance - amountPaid**) . [10 Marks]
- i. The **DepositMachineMain** class represents the application program that instantiates **SavingDeposit** and **LoanPayment** objects, and then invoke their **toString()** method. [6 Marks]

Question 2

Write a complete Java program based on the UML class diagram given in Figure 1 and driver program named `FinalGolf2.java` (refer to Figure 2). Your program should be able to produce the output shown in Figure 3.

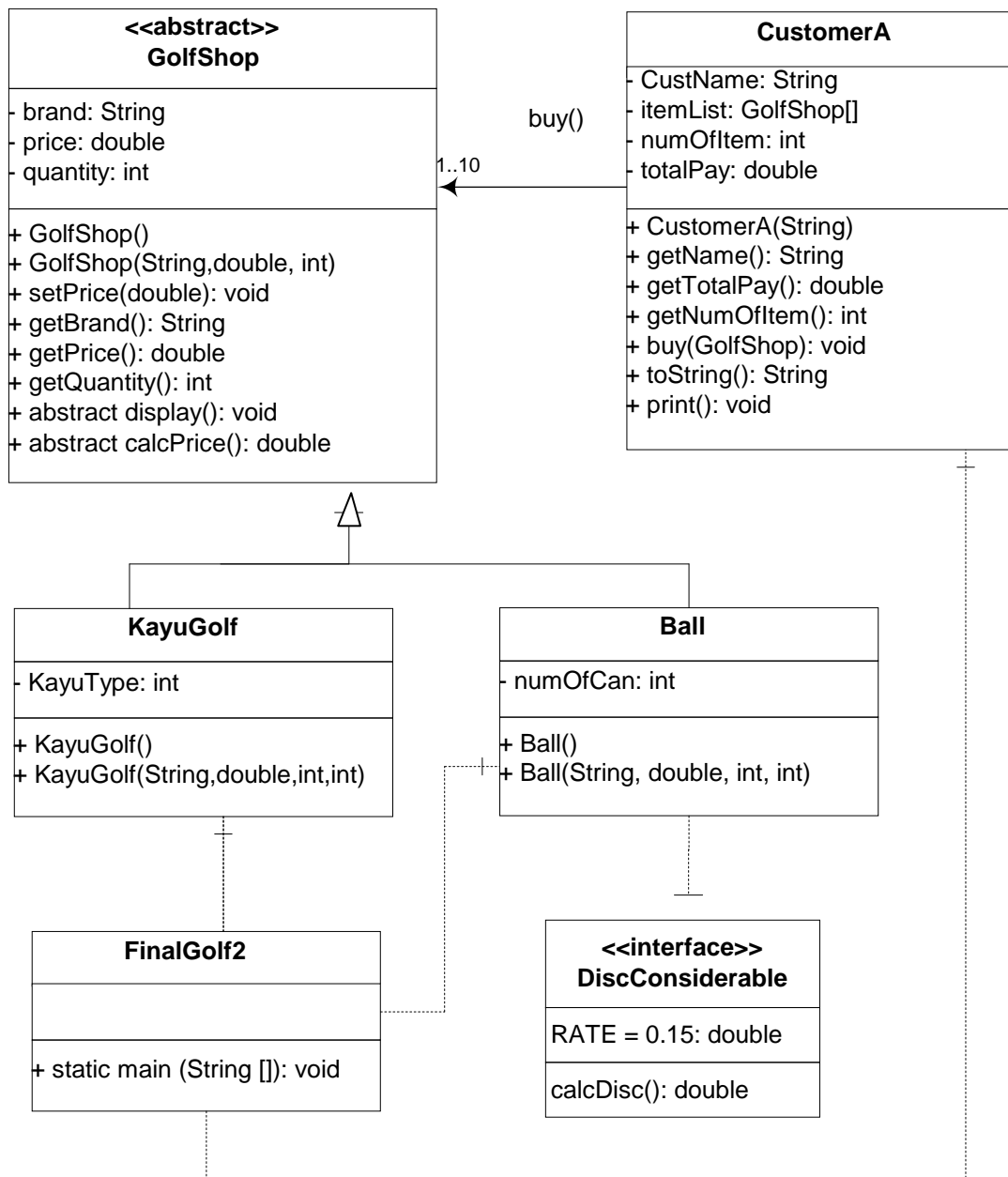


Figure 1: UML class diagram

```

public class FinalGolf2
{
    public static void main(String[] args)
    {
        GolfShop p1 = new KayuGolf("GolfDriver Wood", 0, 3, 2);
        GolfShop p2 = new Ball("Wilson Anne", 225.00, 3, 24);
        GolfShop p3 = new Ball("Cobra", 240.00, 5, 18);
        GolfShop p4 = new KayuGolf("Fairway Wood", 0, 4, 5);

        CustomerA c = new CustomerA("Mikael Nadal");
        System.out.println(c.toString());
        c.buy(p1);
        c.buy(p2);
        c.buy(p3);
        ;
        c.buy(p4);
        ;
        c.print();
        ;
    }
}

```

Figure 2: FinalGolf2.java

You must write five complete Java programs, **GolfShop.java**, **Ball.java**, **KayuGolf.java**, **CustomerA.java** and **DiscConsiderable.java** based on the instruction given below:

- (a) Write an abstract superclass named **GolfShop**. The class provides the following methods:
 - (i) Write the codes for a constructor with no argument that will do nothing.
 - (ii) Write the codes for a constructor with arguments that will initialize all the member attributes to the values received as arguments.
 - (iii) Write suitable codes for the accessor and mutator methods, if applicable.
- (b) Write a subclass **Ball** and implements interface with the following codes:
 - (i) Write the codes for a constructor with no argument that will do nothing.
 - (ii) Write the codes for a constructor with arguments that will initialize all the member attributes for the class, including the superclass's attributes.
 - (iii) Write the codes for **calcDisc()** method that will calculate and return the new price after the discount deducted, that discount received which is 15 percent of the price of the balls.

- (iv) Write the codes for the abstract method **display()** that will print the brand, price, number of can case per set, discount received, and quantity of set of balls can case purchased. The method will also print the total price of set of balls purchased after the discount deducted by calling **calcPrice()**.
- (v) Write the codes for **calcPrice()** that will call **calcDisc()** and return the total price of balls purchased after the discount deducted.
- (c) Write a subclass **KayuGolf** with the following codes:
 - (i) Write the codes for a constructor with no argument that will do nothing.
 - (ii) Write the codes for a constructor with arguments that will initialize all the member attributes for the class, including the superclass's attributes.
 - (iii) Write the codes for the abstract method **display()** that will print the brand, name of racket type, price, and quantity of rackets purchased. Table 1 shows the prices vary according to the type of racket. The method will also print the total price of rackets purchased by calling **calcPrice()**.

Table 1: KayuGolf type and price

Kayu Type	Name of Kayu Golf Type	Price (RM)
1	Driver Wood	80.00
2	Fairway Wood	110.00
3	Iron	235.00
4	Putter	185.00

- (iv) Write the codes for **calcPrice()** that will calculate and return the total price of rackets purchased.
- (d) Write an interface named **DiscConsiderable** with the following codes:
 - (i) Declaration of one final static variable
 - (ii) Declaration of one abstract method
- (e) Write a class named **CustomerA**. The class provides the following methods:
 - (i) Write the codes for a constructor with arguments that accepts the customer's name.
This value should be assigned to the customer's name member variable.
 - (ii) Write the codes for **buy(GolfShop product)** method that will receive one argument of instance, update the array named **itemList** to include the new

- instance of `GolfShop` in the array, update number of tennis items' in the array `itemList`, and print the information of the item bought (using `display()` method) and update the total payment for all items bought (including discount).
- (iii) Write the codes for `toString()` method to return message "*Welcome the customer*" and "*List of Golf Items Bought*".
- (iv) Write the codes for `print()` method that will display message "*TOTAL: RMxx.xx*".

```
WELCOME Mikael Nadal
List of Golf Items Bought

Item 1
KAYUGOLF
Brand   : Yamaha
Type    : Driver wood
Price   : RM110.0
No Discount!
Quantity: 3
Subtotal: RM330.00

Item 2
GOLF BALL
Brand   : Wilson Anne
Price   : RM225.00 for 24 can case
Discount: 15.0%
Quantity: 3
Subtotal: RM573.75

Item 3
GOLF BALL
Brand   : Cobra
Price   : RM240.00 for 18 can case
Discount: 15.0%
Quantity: 5
Subtotal: RM1020.00

Item 4
KAYUGOLF
Brand   : Nike
Type    : Putter
Price   : RM185.0
No Discount!
Quantity: 4
Subtotal: RM740.00
TOTAL: RM2663.75
```

Figure 3: Output of the FinalGolf2.java