Seminar 5 (Classes and Objects) – Exercises (Solution)

1. Write a class Name that models a person’s name. The class has 4 instance variables – gender (char), last name (String), first name(String), middle name(String). It has a constructor that initializes the gender, last, first and middle name. It has get/set accessor methods only for first name , middle name and last name. It has the following other methods:
   a. getFullName() returns the full name. E.g. it may return “Mr. Ong Ah Kow” if the person’s is a male in the sequence last name, first name and middle name.
   b. getInitials() returns the first letter of the first, middle followed by the lastname. E.g. it may return “A. K. Ong”.

```
public class Name
{
   private char gender;
   private String lastName;
   private String firstName;
   private String middleName;

   public Name(String gender, String lName, String fName, String mName)
   {
      this.gender = gender;
      lastName = lname;
      firstName = fname;
      middleName = mName;
   }

   public String getLastName()
   {
      return lastName;
   }

   public void setLastName(String lname)
   {
      lastName = lname;
   }

   //get, set for middle name and first name not done

   public String getFullName()
   {
      if ( gender == ‘M’ || gender ==’m’)
            return “Mr. “+ lastName + “ “ + firstName + “ ” + middleName;
      else
            return “Ms. “+ lastName + “ “ + firstName + “ ” + middleName;
   }

   public String getInitials()
   {
      return firstName.charAt(0) + “. ” + middleName.charAt(0) + “. ”
                  + lastName;
   }
}
```

2. A Rectangle class has 2 instance variables – length (double) and width (double). It has a constructor that initializes the length and width. It also has get/set methods for the length and width. It has several other methods:
    a. getArea() that returns the area of the rectangle
    b. getPerimeter() that returns the perimeter of the rectangle
    c. increaseSize(double length, double width) that increases the length and width of the rectangle.
    d. isBigger(Rectangle r) that returns a boolean value. The method returns true if the current area is bigger than the area of the rectangle passed as parameter. It returns false otherwise.

```
public class Rectangle
{
   private double length;
   private doule width;

   public Rectangle(double length, double width)
   {
      this.length=length;
      this.width = width;
   }

   //get, set methods. Only 1 provided. The rest same
   public String getLength(){ return length; }

   public void setLength(double length) { this.length=length; }

   public double getArea()
   {
      return length * width;
   }

   public double getPerimeter()
   {
      return (length + width) * 2;
   }

   public void increaseSize(double xLen, double xWidth)
   {
      length += xLen;
      width += xWidth;
   }

   public boolean isBigger(Rectangle r)
   {
      if ( this.getArea() > r.getArea() )
         return true;
      return false;
   }
}
```

```
public class Q2
{
   public static void main(String[] args)
   {
      Rectangle r = new Rectangle(10,20);
      System.out.println( "Area is " + r.getArea() );
      System.out.println( "Perimeter is " + r.getPerimeter() );
      Rectangle r1 = new Rectangle(5,10);
      if ( r.isBigger(r1) )
         System.out.println( "First rectangle bigger than second");
      else
         System.out.println("First rectangle is not bigger than second");
   }
}
```

3. [Exam, May 2007]
   James has just bought a plasma television for $5999.90. The brand of the television is Haticho. It has a screen size of 42 inches. It also comes with an extended warranty as an option.
   a. Write a Java class that will model the television. Your class should include four characteristics with distinct data types. It should also include all the usual accessor methods. You need to provide only one constructor for your class.
   b. Demonstrate how you could exercise your class with a simple program.

```
public class Television
{
   private String brand;
   private int size;
   private boolean extendedWarranty;
   private double price;

   public Television(String brand, int size, boolean extendedWarranty,
                                                              double price)
   {
      this.brand = brand;
      this.size=size;
      this.price = price;
      this.extendedWarranty = extendedWarranty;
   }

   //get, set methods. Only 1 provided. The rest same
   public String getBrand(){ return brand; }
   public void setBrand(String brand) { this.brand=brand; }
   public String toString(){
      return brand+" "+size+" "+price+" extended warranty= "+extendedwarranty;
   }
}

public class Q3
{

   public static void main(String[] args)
   {
      Television tv = new Television("Hitachi", 48, false, 2500);
      tv.setExtendedWarranty(true);
      System.out.println( tv );
   }
}
```

4a. A BankAccount class has two instance variables: accountId and balance. It has a constructor that has 2 arguments to initialize the accountId and the balance. It has get methods for accountId and balance. It has a deposit method that accepts an argument that represents the amount to deposit. The method adds the amount to the balance. It has a withdraw method that accepts an argument which represents the withdrawal amount. This method returns a boolean value. If the withdrawal amount is greater than the balance, no withdrawal is made and a false value is returned. Otherwise, the amount is deducted from the balance and a true value is returned. It also has a transfer method that accepts 2 arguments – another bank account to transfer to and the amount to transfer. The toString method returns the accountId and balance as a string. Write the BankAccount class.

4b. Write statements for the following:

i. The BankAccount class.
ii. Create a Bank Account object for id “1001” with an initial balance of 100 dollars.
iii. Call the toString method to print the details of this object.
iv. Deposit 50 dollars to this account.
v. Withdraw 100 dollars from this account and print if the withdrawal is successful.
vi. Repeat step e by withdrawing another 100 dollars and print the outcome.

```
public class BankAccount{
   private String accountId;
   pricate double balance;

   public BankAccount(String id, double balance){
      accountId = id;
      this.balance = balance;
   }

   //get, set methods not shown

   public void deposit(double amount){
      balance += amount;
   }

   public boolean withdraw(double amount){
      if ( amount <= balance){
            balance -= amount;
                 return true;
            }
            return false;
   }

   public void transfer(BankAccount ba, double amount){
      ba.deposit(amount);
      balance -= amount;
   }

   public String toString(){
      return accountId + “ ” + balance;
   }

}
```

```
public class Q4
{
   public static void main(String[] args)
   {
      BankAccount ba = new BankAccount(“1001”, 100);
      System.out.println(ba);
      ba.deposit(50);
      if ( ba.withdraw(100))
            System.out.println(“Withdraw successful”);
      else
            System.out.println(“Not successful”);

   }
```

5. Write a Phone class that encapsulates speed dialing feature. Speed dialing allows one of the digits (0-9) to be assigned a phone number so that the digit can be used to quickly dial a number. The Phone class consists of the following:

   a. Instance variables:
      i. It has an instance variable representing the phone number (int).
      ii. It has another instance variable – an int array to store numbers for speed dialing.

   b. Constructor
      The constructor has a parameter that initializes it's phone number. The integer array of size 10 is also created here. The index of the array will be the speed digit and the element will store the phone number. Initially all the elements in the array will be 0.

   c. It has the following methods:
      i. void assignSpeedDial(int digit, int number)
         This method assigns a digit (must be 0-9) with a number. Display "invalid digit" if the digit is not between 0-9.
      ii. void speedDial(int digit)
         This method dials a number based on the digit parameter (must be 0-9). If the digit has been assigned with a number (i.e. not zero), then display "calling 999999", where 999999 is the number assigned to the digit. If the digit is not assigned with a number, display "No number assigned".
      iii. void displayAllSpeedDial()
         This method displays the numbers assign to each digit(0-9). A sample is as follows:

```
0 – not assigned
1 – 91232456
2 – 81234567
3 – not assigned
...
9 – 82468124
```

Test the Phone class.

```
public class Phone{
   private int number;
   pricate int[] dial;

   public Phone(int number){
      this.number = number;
      dial = new [10];
   }

   public void assignSpeedDial(int digit, int number)
   {
      if ( digit >=0 && digit < 10)
         dial[digit]=number;
   }

   public void speedDial( int digit)
   {
      if ( digit >=0 && digit <10)
         if ( dial[digit] ==0)
            System.out.println("Not assigned");
         else
            System.out.println("Dialling... "+dial[digit]);
   }

   public void displayAllSpeedDial()
   {
      for (int n=0; n<10; n++)
         if ( dial[n]==0)
            System.out.println(n + " - not assigned");
         else
            System.out.println(n + " - " + dial[n]);

   }
}

public class Test
{
   public static void main(String[] args)
   {
      Phone p = new Phone(91231332);
      p.asignSpeedDial(0, 11111111);
      p.assignSpeedDial(10, 23234234);
      p.speedDial(1);
      p.dislayAllSpeedDial();
   }

}
```