**Days of the Future Past**

**Due – 25th January , 2015**

1. Create a class called Square as described below:

* Fields:   
  height, width
* Methods:  
  public double getHeight()  
  public void setHeight(double h)  
  public double getWidth ()  
  public void setWidth (double w)  
  public double getArea ()

Hint: If I take your class and use it following would be the code and the output.

|  |  |
| --- | --- |
| Code | Output |
| double h, w, a;  Square s1 = new Square();  s1.setHeight(3);  s1.setWidth(4);  h = s1.getHeight();  w = s1.getWidth();  a = s1.getArea();  System.out.println(“Height = ”+ h);  System.out.println(“Width = ”+ w);  System.out.println(“Area = ”+ a); | Height = 3.0  Width = 4.0  Area = 12.0 |

1. Design a “Vehicle” class. A vehicle assumes that the whole world is a 2 dimensional graph paper. It maintains its x and y coordinates (both are integers). The vehicle gets manufactured (constructed?) at (0,0) coordinate.Write a user class called “Vehicle”. It must have methods to move up, down, left, right and a toString method for printing current coordinate.

Note: All moves are 1 step. That means a single call to any move method changes value of either x or y or both by 1.

Take help from:

http://www.javabeginner.com/learn-java/java-tostring-method

http://cscie160-distance.com/toString.html

|  |
| --- |
| public class VehicleUser{  public static void main(String[] args){  Vehicle car = new Vehicle();  System.out.println(car.toString());  car.moveUp();  System.out.println(car.toString());  car.moveLeft();  System.out.println(car.toString());  car.moveDown();  System.out.println(car.toString());  car.moveRight();  // see, output for following two lines are same because toString() is automatically called. So, you can omit toString when printing.  System.out.println(car.toString());  System.out.println(car);  }  } |
| Expected Output:  (0, 0)  (0, 1)  (-1, 1)  (-1, 0)  (0, 0)  (0, 0) |

1. Create a class called BankAccount as described below:

* **Fields:**  
  name, address, accountID, balance
* **Methods:**  
  public String getName()  
  public void setName(String n)  
  public String getAccountID()  
  public void setAccountID(String i)  
  public String getAddress()  
  public void setAddress(String a)  
  public double getBalance()  
  public void setBalance(double c)  
  public void addInterest() //adds 7% of the balance

Write a class called Main to write a main() method:

* public static void main(String[] args){  
    
  }
* Inside the main() method
  + Create 3 objects/instances of BankAccount called acc1, acc2 and acc3
  + Set their fields to some value using the public methods.
  + Call addInterest() on acc1 and acc3
  + Print the information of each BankAccount using System.out.println()

Add constructors to Student and BankAccount and use the constructor to set the field values.

1. Ask user for dimension/size of two row matrices, take all values in each matrix and print the summation. If matrices are A and B, then you need to calculate C=5A-B. For example, if the user enters 3, then size of each matrix is 3. Then you need to take 3 values for first matrix which will form A=[5 6 7]. Then take 3 values for second matrix, B=[2 3 4]. You need to print result, C=[23 27 31]. The output will be 23 27 31. You must use array, loop (while or for), and array.length for this program.
2. Create a class called Date that includes three pieces of information as instance variables a month (type int), a day (type int) and a year (type int). Your class should have a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a set and a get method for each instance variable. Provide a method displayDate that displays the month, day and year separated by forward slashes (/). Write a test application named DateTest that demonstrates class Date's capabilities.

1. Design a ComplexNumber class.
   1. Take a ComplexNumber from user.
   2. Implement increaseValue() to increase both real & imaginary part by 2.
   3. Print the ComplexNumber from main method using toString.
   4. Print polar radius from main method using print( a.polarRadius() ). Hint: If a complex number is x+iy, then its polar radius is
2. Design a point class
   1. Take value of point from the user.
   2. create a method to calculate distanceFromOrigin
   3. print the distance from main method
3. Design a class for Quadratic equation named QuadEqn consisting of 3 coefficients, a,b,c
   1. Take input from user in main
   2. print this QuadEqn
   3. create a method to calculate 1st root
   4. print one of the roots from main. (assume all roots are real, use + instead of +/-)

For example, if user gives 1, -5, 6

then print the equation: x2 - 5x + 6

and 3 (the root/solution)

1. Array Left Rotate and Right Rotate Consider the 1,4,8,16,25,36,49,64,81,100. Now Right and left rotate the array by the no. of position given by the user. Example: If user gives ‘4’ as no. of position to be rotated, your left rotate should give output ‘25,36,49,64,81,100, 1,4,8,16’ and right rotate should give output ‘49,64,81,100, 1,4,8,16,25,36’.
2. Write the **Mango** and the **Jackfruit** classes so that the following code generates the output below:

|  |
| --- |
| **public class Test{** |
| **public static void testFruit(Fruit f){** |
| **System.out.println("----Printing Detail-----");** |
| **if(f.hasFormalin()){** |
| **System.out.println("Do not eat the "+f.getName()+".");** |
| **System.out.println(f);** |
| **}else{** |
| **System.out.println("Eat the "+f.getName()+".");** |
| **System.out.println(f);** |
| **}** |
| **}** |
| **public static void main(String [] args){** |
| **Mango m = new Mango();** |
| **testFruit(m);** |
| **Jackfruit j = new Jackfruit();** |
| **testFruit(j);** |
| **}** |
| **}** |
| **public class Fruit{** |
| **private boolean formalin = false;** |
| **public String name = "";** |
| **public Fruit(boolean formalin, String name){** |
| **this.formalin = formalin;** |
| **this.name = name;** |
| **}** |
| **public String getName(){** |
| **return name;** |
| **}** |
| **public boolean hasFormalin(){** |
| **return formalin;** |
| **}** |
| **}** |

**----Printing Detail-----**

**Do not eat the Mango.**

**Mangos are bad for you**

**----Printing Detail-----**

**Eat the Jackfruit.**

**Jackfruits are good for you**

1. Write the **ComplexNumber** class so that the following code generates the output below:

|  |
| --- |
| **public class Tester {** |
| **public static void main(String[] args) {** |
| **RealNumber rn = new ComplexNumber();** |
| **System.out.println(rn);** |
|  |
| **System.out.println("--------------------");** |
|  |
| **rn = new ComplexNumber(5, 7);** |
| **System.out.println(rn);** |
|  |
| **System.out.println("--------------------");** |
| **ComplexNumber cn = new ComplexNumber();** |
| **cn.check();** |
| **}** |
| **}** |
| **public class RealNumber {** |
| **private double realValue;** |
| **public double getRealValue() {** |
| **return realValue;** |
| **}** |
| **public void setRealValue(double r) {** |
| **realValue = r;** |
| **}** |
| **public RealNumber() {** |
| **this(0);** |
| **}** |
| **public RealNumber(double r) {** |
| **setRealValue(r);** |
| **}** |
| **public String toString() {** |
| **return "RealPart: "+getRealValue();** |
| **}** |
| **public void ping() {** |
| **System.out.println("I'm in RealNumber class");** |
| **}** |
| **}** |

**RealPart: 1.0**

**ImaginaryPart: 1.0**

**--------------------**

**RealPart: 5.0**

**ImaginaryPart: 7.0**

**--------------------**

**I'm in ComplexNumber class**

**I'm in RealNumber class**

**Checking ended.**

1. Write the **FootBallTeam and the CricketTeam** classes so that the following code generates the output below:

**public class Sports{**

**public static void printDetail(Team t){**

**System.out.println("----Printing Detail-----");**

**System.out.println(t);**

**}**

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

**FootBallTeam f = new FootBallTeam("Brazil");**

**CricketTeam c = new CricketTeam("India");**

**printDetail(f);**

**printDetail(c);**

**}**

**}**

**public class Team{**

**protected String name = null;**

**Team(String name){**

**this.name = name;**

**}**

**}**

**----Printing Detail-----**

**Our name is Brazil**

**we play Football**

**----Printing Detail-----**

**Our name is India**

**we play Cricket**

1. Write the CheckingAccount class so that the following code generates the output below

|  |
| --- |
| **public class Account{** |
| **protected double balance = 0.0;** |
| **public Account(double balance){** |
| **this.balance = balance;** |
| **}** |
| **public double getBalance(){** |
| **return balance;** |
| **}** |
| **}** |
| **public class TestAccount{** |
| **public static void printBalance(Account a){** |
| **System.out.println("Account Balance: " + a.getBalance());** |
| **}** |
| **public static void main(String [] args)** |
| **{** |
| **System.out.println("Number of Checking Accounts: " + CheckingAccount.numberOfAccount);** |
| **printBalance(new CheckingAccount());** |
| **printBalance(new CheckingAccount(100.00));** |
| **printBalance(new CheckingAccount(200.00));** |
| **System.out.println("Number of Checking Accounts: " + CheckingAccount.numberOfAccount);** |
| **}** |
| **}** |

**Number of Checking Accounts: 0**

**Account Balance: 0.0**

**Account Balance: 100.0**

**Account Balance: 200.0**

**Number of Checking Accounts: 3**

1. Write the **CSEStudent** and **CSE111Student** class so that the following code generates the output below **[Answer on the answer-script]**

|  |
| --- |
| **public class Student{** |
| **public String msg = "I love BU";** |
| **public String shout(){** |
| **return msg;** |
| **}** |
| **}** |
| **public class TestStudent{** |
| **public static void printShout(Student s){** |
| **System.out.println("------------------");** |
| **System.out.println(s.msg);** |
| **System.out.println(s.shout());** |
| **}** |
| **public static void main(String [] args){** |
| **Student s = new Student();** |
| **CSEStudent cs = new CSEStudent();** |
| **CSE111Student cs111 = new CSE111Student();** |
| **System.out.println(s.msg);** |
| **System.out.println(cs.msg);** |
| **System.out.println(cs111.msg);** |
| **printShout(s);** |
| **printShout(cs);** |
| **printShout(cs111);** |
| **}** |
| **}** |

|  |
| --- |
| **Output** |
| **I love BU**  **I want to transfer to CSE**  **I love Java Programming**  **------------------**  **I love BU**  **I love BU**  **------------------**  **I love BU**  **I want to transfer to CSE**  **------------------**  **I love BU**  **I love Java Programming** |

1. Write the **removeOdd** function bellow which take in an array of numbers that has even and odd numbers mixed. This function **removes** the odd numbers and returns a **compact** array which only has the even numbers. For example output of the following code is:

**21 33 44 66 11 1 88 45 10 9**

**44 66 88 10**

**public class Test{**

**public static int [] removeOdd (int [] input){**

**//Your code here**

**}**

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

**int [] mixedArray = {21, 33, 44, 66, 11, 1, 88, 45, 10, 9};**

**for (int i = 0; i < mixedArray.length; i++) {**

**System.out.print(mixedArray[i] + " ");**

**}**

**System.out.println();**

**int [] noOdd = removeOdd(mixedArray);**

**for (int i = 0; i < noOdd.length; i++) {**

**System.out.print(noOdd[i] + " ");**

**}**

**}**

**}**

**-The End -**