Assignments

Q1. Accept a integer number and when the program is executed print the binary, octal and hexadecimal equivalent of the given number.
Sample Output:
java Test
Enter Number : 20
Given Number :20
Binary equivalent :10100
Octal equivalent :24
Hexadecimal equivalent :14
Hint : Use Wrapper Class (toBinaryString() , toOctalString(), toHexString())

Q2. Accept 2 double values from User (using Scanner). Check data type. If arguments are not doubles, supply suitable error message & terminate.
If numbers are double values, print its average.
Hint : Check Scanner class for the methods

Q3. Display food menu to user. User will select items from menu along with the quantity. (eg 1. Dosa 2. Samosa 3. Idli ... 10 . Generate Bill) Assign fixed prices to food items(hard code the prices) When user enters 'Generate Bill' option , display total bill & exit.
(No need of any class, and no need to display the food details. Only display the total of orderd food)

Q4. Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as fields—a part number (type String), a part description (type String), a quantity of the item being purchased (type int) and a price per item (double). Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable. calculates the invoice amount (i.e. multiplies the quantity by the price per item), then returns the amount as a double value.
If the quantity is not positive, it should be set to 0.
If the price per item is not positive, it should be set to 0.0.
Write a test application named InvoiceTest that demonstrate class Invoice's capabilities.

Q5. Create a class Point2D , in package - "com.app.geometry" : for representing a point in x-y co-ordinate system. Create a parameterized constructor to init x & y co-ords. Add a method to return string form of point's x & y co-ords
Hint : public String getDetails())

Add isEqual method to Point2D class :a boolean returning method : must return true if n only if both points are having same x,y co-ords or false otherwise.
Add calculateDistance method to calculate distance between current point and specified point & return the distance to the caller.

Hint : Use distance formula . Use java.lang.Math class methods --sqrt, pow etc.
Write TestPoint class , in package "tester" , with a main method, Accept co ordinates of 2 points from user (Scanner) --to create 2 points (p1 & p2)

Assignments

Use getDetails method to display point details.(p1's details & p2's details)

Invoke isEqual & display if points are same or different (i.e p1 & p2 are located at the same position)
If they are not located at the same position , display distance between p1 & p2

Q6 Apply inheritance n polymorphism
a) Arrange Fruit,Apple,Orange,Mango in inheritance hierarchy
b) Properties (instance variables) : color : String , weight : double , name:String, isFresh : boolean
c) Add suitable constructors.
d) Override toString correctly to return state of all fruits (including : name ,color , weight)
e) Add a taste() method : public String taste()

For Fruit : it should return "no specific taste"
Apple : should return "sweet n sour"
Mango : should return "sweet"
Orange : should return "sour"

f) Add all of above classes under the package "com.app.fruits"
g) Create a Class FruitBasket , with main method inside it. Use it for testing
h) Prompt user for the basket size n create suitable data structure and give options

0. Exit

1. Add Mango
case 1 : boundary checking
basket[counter++]=new Mango(nm,weight,color);
break;

2. Add Orange

3. Add Apple

NOTE : You will be adding a fresh fruit in the basket , in all of above options.

4. Display names of all fruits in the basket.
eg : for-each --- null checking --getName()

5. Display name,color,weight , taste of all fresh fruits , in the basket.
eg : for-each , null checking --toString , taste, isFresh : getter

6. Display tastes of all stale(not fresh) fruits in the basket.

7. Mark a fruit as stale
i/p : index
eg : setter : isFresh : false
o/p : error message (in case of invalid index) or mark it stale

Assignments

8. Mark all sour fruits stale (optional)
eg : for-each , taste --equals(String)

Q7. Define a new exception, called ExceptionLineTooLong, that prints out the error message "The strings is too long". Write a program that reads a String from user and calculates its length. and throws an exception of type ExceptionLineTooLong in the case where a string of length is more than 80 characters.
(Hint- Use String class length() method)

Q8. Store book details in a library in hashing based data structure : HashSet
Book details : isbn(string), category(enum), price(double), authorName(string), quantity(int) .
Unique ID : isbn

Write a menu driven code that have below menus.
1. Accept book details and add in collection
2. Display all books from collection
3. Sort the book details as per category and display it.
4. Sort the book details as per author and display it.
5. Find book by id.

Q9. Write a Java program to check if a file or directory specified by pathname exists or no.