

Java Internship Assignments

Day01

Assignment - 1

Q1. Write a program for accepting day, month and year values. Declare a class Date having fields day, month, year.

Implement the following methods.

- 1) initDate();
- 2) printDate();
- 3) acceptDate();

Q2. Write a menu driven program for accepting Student info.

Create a class Student with data members roll no, name and marks.

Implement the following methods:

- 1) Initialize Student object with default values. (Hint: parameterless constructor)
- 2) void printStudentDetails()
- 3) void acceptStudentDetails()

Q3. Write a class Cylinder to find volume of a Cylinder by using fields radius, height.

(volume of Cylinder = $3.14 * \text{radius} * \text{radius} * \text{height}$)

Class having following methods:

Cylinder(), Cylinder(double radius, double height), getRadius(), setRadius(), getHeight(), setHeight(), getVolume(), printVolume()

Initialize data members using constructor member initializer list.

Q4. 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 ordered food)

Q5. Implement Q1 & Q2 two questions using Menu Driven program. (Hint: Switch Case)

OR

Q1. 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 java application FruitBasket , with main method , as a tester

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.