Book Detail

Create a class Book with the following private member variables

- String bookName
- int bookPrice
- String authorName

Include appropriate getters and setters method.

Create a class TestBook which has the main method. Get the details as shown in the sample input. Create an object for book class and assign the value for its attrbutes using the setters. Print the output as shown in the sample output using the getters method.

Note: Use the same attribute names as given in the question and camel case notation for methods. Name of book and author can have space in between.

Sample Input 1:

```
Enter the Book name:
Java
Enter the price:
500
Enter the Author name:
Einstein
```

Sample Output 1:

```
Book Details
Book Name :Java
Book Price :500
Author Name :Einstein
```

Student Details - Constructor

Create a class Student with the private attributes

int studentId

String studentName, studentAddress, collegeName.

Include appropriate getter methods.

Assume most of the students are from “NIT” college. So user has to give input whether the student is from NIT or not.

1. If student belongs to NIT, give input as 'yes/YES' and skip input for the attribute collegeName and create student object with 3-argument constructor to initilze the values for studentId, studentName and studentAddress and collegeName as “NIT”.
2. If student belongs to other college, give input as 'no/NO' and get college name from the user and create student object with 4-argument constructor to initialize all the values.
3. Instead of Yes / No, if user enters different input then display 'Wrong Input' and get the input again.

Based on the above assumptions write the necessary constructors in the Student class.

Write a class StudentMain with the main method and test the application.

Sample Input 1:

Enter Student's Id:

12

Enter Student's Name:

John

Enter Student's address:

Chennai

Whether the student is from NIT(Yes/No):

NO

Enter the college name:

SVS

Sample Output 1:

Student id:12

Student name:John

Address:Chennai

College name:SVS

Sample Input 2:

Enter Student's Id:

43

Enter Student's Name:

Tom

Enter Student's address:

Coimbatore

Whether the student is from NIT(Yes/No):

y

Wrong Input

Whether the student is from NIT(Yes/No):

yes

Sample Output 2:

Student id:43

Student name:Tom

Address:Coimbatore

College name:NIT

Volume calculator- Over Loading

Sarah got confused to calculate volume of cylinder and cuboid. Write a Java application to help Sarah to do this.

Create a class called VolumeCalculator that has the following methods
double calculateVolume(double radius,double height)
This method calculates the volume of the cylinder using the formula 3.14*radius*radius*height
double calculateVolume(int length,int breadth,int height)
This method calculates the volume of the cuboid using the formula length*breadth*height
Write a TestMain class to test the application.

Sample Input

Enter the choice
1.Find Volume For Cylinder
2.Find Volume For Cuboid
1
Enter the radius
3
Enter the height
2

Output
Volume of the Cylinder is 56.52

Sample Input

Enter the choice
1.Find Volume For Cylinder
2.Find Volume For Cuboid
2

Enter the length
3
Enter the breadth
2
Enter the height
1

Output

Volume of the Cuboid is 6.00

Ticket Price Calculation - Static

Ticket Calculation

Create a class Ticket with the following private variables
int ticketid;
int price;
static int availableTickets;

Include getters and setters methods in the Ticket class.

AvailableTickets should hold only positive value. Zero and negative values are not allowed.(This logic should be checked inside the corresponding setter method)

Write the following method in the Ticket class:

public int calculateTicketCost(int nooftickets) –this method should check the ticket availability, If the tickets are available, reduce the nooftickets from availableTickets and calculate the total amount as nooftickets*price and return the total amount. If the tickets are not available, this method should return -1.

Write a main method in the Main class to test the application.

Sample input and output

```
Enter no of bookings:
2
Enter the available tickets:
25
Enter the ticketid:
123
Enter the price:
100
Enter the no of tickets:
5
Available tickets: 25

Total amount:500

Available ticket after booking:20

Enter the ticketid:
124
Enter the price:
100
Enter the no of tickets:
2
Available tickets: 20
```

```
Total amount:200

Available ticket after booking:18
```

Sample input and output 2:

```
Enter no of bookings:
1
Enter the available tickets:
25
Enter the ticketid:
123
Enter the price:
100
Enter the no of tickets:
26
Tickets not sufficient / available
```

Employee Salary Calculation

Create a class Employee with the following private member variables.

- int employeeId
- String employeeName
- double salary
- double netSalary

Include appropriate getters and setters method in Employee class. Write the following method in the Employee class:
public void calculateNetSalary(int pfpercentage) - This method should take PF percentage as argument. Deduct the PF amount from the salary and set the netSalary.

Create a Main class which has the main method which invokes the method to get the input and prints the details as shown in the sample.

Also write a method :

public static Employee getEmployeeDetails() - which gets the employee details and returns the employee object.

public static int getPFPercentage() - which gets the PF percentage and returns the same

Sample Input 1:

```
Enter Id:
101
Enter Name:
Vivek
Enter salary:
20000
Enter PF percentage:
7
```

Sample Output 1:

```
Id : 101

Name : Vivek

Salary : 20000.0

Net Salary : 18600.0
```

Travel Details

Create a class Person with the following private member variables

- String name
- char gender
- int age

Include appropriate getters and setters method
Create a class BusTicket with the following private member variables

- int ticketNo
- float ticketPrice
- float totalAmount
- Person person

Include appropriate getters and setters method

Write the following method in the BusTicket class
void calculateTotal()—this method should calculate the total and set it based on the discount given below:

1. For Children whose age is less than 16, give 50% discount
2. For Senior citizen whose age is greater than or equal to 60 give 25% discount
3. For Ladies give 10% discount
4. No discount for remaining category.

Based on above condition calculate total price.
Create TestMain class which has main method and do the following input and output process .

To get the input write a method

public static BusTicket getTicketDetails() - Get the inputs relevant to BusTicket in this method and return the BusTicket object. Call this method in the main method, using the returned object invoke the method calculateTotal and print the output accordingly

Sample input 1:

Enter the passenger name:
vivek
Enter the gender(M or F / m or f):
M
Enter the age:
16
Enter the ticket no:

123
Enter the ticket price:
100.0

Sample Output 1

Ticket no:123
Passenger Name:vivek
Price of a ticket : 100.0
Total Amount : 100.0

Sample input 2:
Enter the passenger name:
Priya
Enter the gender(M or F / m or f):
F
Enter the age:
61
Enter the ticket no:
140
Enter the ticket price:
500.0

Sample Output 1

Ticket no:143
Passenger Name:Priya
Price of a ticket : 500.0
Total Amount : 375.0

BankAccountDetails

In the first round of HR interview for a banking sector, HR decides to make candidates design an application which provides only information on transaction like amount withdrawn with respect to fields given. Develop a program to implement this scenario.

Create a class Account with the attributes:

- accountId int
- accountType String
- balance int

The method **boolean withdraw(int)** used to calculate the current balance of the respective account. Before that it should enough balance. If there is enough balance, deduct the amount from the balance and print "Balance amount after withdraw: XXX" and return true. If there is no enough balance, print "Sorry!!! No enough balance" and return false.

Create a class AccountDetails with main function and the below methods :

- public Account getAccountDetails() - This methods gets the input related to Account from the user and returns the Account object with all values set. If the input given for balance is less than or equal to zero, consider it as invalid and display "Balance should be positive". Continue this kind of evaluation till user enters a positive value.

- public int getWithdrawAmount() - This methods gets the amount to be withdrawn as input from the user and returns the same. If the input given for amount is less than or equal to zero, consider it as invalid and display "Amount should be positive". Continue this kind of evaluation till user enters a positive value.

Use appropriate getters and setters.

Sample input 1:

Enter account id:

100
Enter account type:

Savings

Enter balance:

10000
Enter amount to be withdrawn:

500

Sample Output 1:

Balance amount after withdraw: 9500

Sample input 2:
Enter account id:

101
Enter account type:

Savings
Enter balance:
1000
Enter amount to be withdrawn:
1500

Sample Output 2:

Sorry!!! No enough balance

Sample input 3:

Enter account id:

100

Enter account type:

Savings

Enter balance:

-100

Balance should be positive

Enter balance:

5000

```
Enter amount to be withdrawn:
500
```

Sample Output 1:

```
Balance amount after withdraw: 4500
```

Movie Ticket Calculation

Create a class Movie with the following private member variables.

- String movieName
- String movieCategory
- int ticketCost

Include appropriate getters and setters method in Movie class.
Write the following method in the Movie class:
public int calculateTicketCost(String circle)--This method should take circle as argument. Return 0 when any one of the following condition met :

- If circle is gold and movie category is 2D set ticket cost Rs.300
- If circle is gold and movie category is 3D set ticket cost Rs.500
- If circle is silver and movie category is 2D set ticket cost Rs.250
- If circle is silver and movie category is 3D set ticket cost Rs.450

Return -1 if category is invalid. Return -2 if circle is invalid and -3 if both are invalid.

Create a Main class to get the input and to display the output

Get the input for Movie using the method - public static Movie getMovieDetails(). Invoke this method from the main method and store the Movie object.

Get the input for circle using the method - public static String getCircle(). Invoke this method from the main method and store the circle value.

Next, invoke the calculateTicketCost method in Movie by passing this circle and calculate the ticket cost and set to the ticketCost attribute and print the output.

If the return from calculateTicketCost is zero, the output should be be "The ticket cost is=<ticket cost>".

If it returns -1, the output should be "Sorry there is no <category> type of category in theater."

If the return is -2, the output should be "Sorry!!! Circle is Invalid."

If the return is -3, the output should be "Sorry!!! Both circle and category are Invalid."

Sample Input 1:

Enter the movie name:
VIP

```
Enter the movie category:
2D
Enter the circle:
gold
```

Sample Output 1:

```
Movie name = VIP
Movie category = 2D
The ticket cost is = 300
```

Sample Input 2:

```
Enter the movie name:
Titanic
Enter the movie category:
5D
Enter the circle:
GOLD
```

Sample Output 2:

```
Movie name = Titanic
Movie category = 5D
Sorry there is no 5D type of category in theater.
```

Sample Input 3:

```
Enter the movie name:
Black Panther
Enter the movie category:
2d
Enter the circle:
SILVER
```

Sample Output 3:

```
Movie name = Black Panther
Movie category = 2d
The ticket cost is = 250
```

Sample Input 4:

```
Enter the movie name:
Black Panther
Enter the movie category:
2d
Enter the circle:
good
```

Sample Output 4:

```
Movie name = Black Panther
Movie category = 2d
Sorry!!! Circle is Invalid.
```

Average and Grade Calculation

Develop a smart application as Student Grade Calculator(SGC).

Create a class Student with following private attribute :

int id, String name, marks(integer array), float average and char grade. Include appropriate getters and setters methods and constructor.

void calculateAvg()- This method should calculate average and set average mark for the current student.

void findGrade()- This method should set the grade based on the average calculated. If the average is between 80 and 100 then, then return grade as 'O', else 'A' .If the student gets less than 50 in any of the subjects then return grade as 'F'. Using appropriate setter method set the grade to the student.

(Note : number of subject should be greater than zero, if not display as 'Invalid number of subject' and get number of subject again, Assume mark for a subject should be in the range 0 - 100. If not display a message "Invalid Mark" and get the mark again)

Write a class StudentMain and write the main method.

In this class, write a method

public static Student getStudentDetails() - this method should get the input from the user for a student, create a student object with those details and return that object.

In main create student's object by invoking the getStudentDetails method. Also calculate average and grade for that student object using appropriate methods.

SGC app should get the input and display the output as specified in the snapshot:

Sample Input 1:

```
Enter the id:
123
Enter the name:
Tom
Enter the no of subjects:
3
Enter mark for subject 1:
95
Enter mark for subject 2:
80
```

```
Enter mark for subject 3:
75
```

Sample Output 1:

```
Id:123
Name:Tom
Average:83.33
Grade:O
```

Sample Input 2:

```
Enter the id:
123
Enter the name:
Tom
Enter the no of subjects:
0

Invalid number of subject

Enter the no of subjects:

3
Enter mark for subject 1:
75
Enter mark for subject 2:
49
Enter mark for subject 3:
90
```

Sample Output 2:

```
Id:123
Name:Tom
Average:71.33
Grade:F
```

Student and Department Detail

Create a class Department with the following private member variables

- int did
- String dname

Include appropriate getters and setters method in Department class.

Create a class Student with the following private member variables

- int sid
- String sname
- Department department

Include appropriate getters and setters method in Student class.

Create a TestMain class which has main method.

In addition to main method, create a method

public static Student createStudent() - All input as shown in the sameple input should be got in this method. Set the values to the Student object and return that object

Note : In main method, invoke the createStudent method and print the details of the object returned by that method.

Sample Input 1:

```
Enter the Department id:
100
Enter the Department name:
Computerscience

Enter the Student id:
123
Enter the Student name:
sudha
```

Sample Output 1:

```
Department id:100
Department name:Computerscience
Student id:123
Student name:sudha
```