1) Write a Program to accept two integers through the command line
argument and print the sum of the two numbers

Example:

```
C:\>java Sample 10 20
```

O/P Expected : The sum of 10 and 20 is 30

Write a Program to accept two integers through the command line
argument and print the sum of the two numbers

Example:

```
C:\>java Sample 10 20
```

O/P Expected : The sum of 10 and 20 is 30

2) Write a Program that prints prime numbers from 1 to n

3) Write a program to print month in words, based on input month in numbers
Example1:

```
C:\>java Sample 12
```

O/P Expected : December

Example2:

```
C:\>java Sample
```

O/P Expected : Please enter the month in numbers

Example3:

```
C:\>java Sample 15
```

O/P Expected : Invalid month

4) Write a program to print the sum of the elements of the array with the given below condition. If the array has 6 and 7 in succeeding orders, ignore 6 and 7 and the numbers between them for the calculation of sum.
Eg1) Array Elements - 10,3,6,1,2,7,9
O/P: 22
[i.e 10+3+9]
Eg2) Array Elements - 7,1,2,3,6
O/P:19
Eg3) Array Elements - 1,6,4,7,9
O/P:10

5) Create a new class called “Calculator” which contains the following:
1. A static method called powerInt(int num1,int num2) that accepts two integers and returns num1 to the power of num2 (num1 power num2).
2. A static method called powerDouble(double num1,int num2) that accepts one double and one integer and returns num1 to the power of num2 (num1 power num2).

3. Call your method from another class without instantiating the class (i.e. call it like Calculator.powerInt(12,10) since your methods are defined to be static)
Hint: Use Math.pow(double,double) to calculate the power.

6) CCreate a class called Author is designed as follows:

It contains:

- Three private instance variables: name (String), email (String), and gender (char of either 'm' or 'f').
- One constructor to initialize the name, email and gender with the given values.

And, a class called Book is designed as follows:
It contains:

- Four private instance variables: name (String), author (of the class Author you have just created), price (double), and qtyInStock (int). Assuming that each book is written by one author.
- One constructor which constructs an instance with the values given.
- Getters and setters: getName(), getAuthor(), getPrice(), setPrice(), getQtyInStock(), setQtyInStock(). Again there is no setter for name and author.

Write the class Book (which uses the Author class written earlier).
Try:
1. Printing the book name, price and qtyInStock from a Book instance. (Hint: aBook.getName())
2. After obtaining the "Author" object, print the Author (name, email & gender) of the book.

7) Create a class called Person with a member variable name. Save it in a file called Person.java Create a class called Employee who will inherit the Person class.The other data members of the employee class are annual salary (double), the year the employee started to work, and the national insurance number which is a String.Save this in a file called Employee.java
Your class should have a reasonable number of constructors and accessor methods. Write another class called TestEmployee, containing a main method to fully test your class definition.

8) Write a program to create a class named shape. It should contain 2 methods- draw() and erase() which should print "Drawing Shape" and "Erasing Shape" respectively.
For this class we have three sub classes- Circle, Triangle and Square and each class override the parent class functions- draw () and erase ().
The draw() method should print "Drawing Circle", "Drawing Triangle", "Drawing Square" respectively.
The erase() method should print "Erasing Circle", "Erasing Triangle", "Erasing Square" respectively.
Create objects of Circle, Triangle and Square in the following way and observe the polymorphic nature of the class by calling draw() and erase() method using each object.
Shape c=new Circle();
Shape t=new Triangle();
Shape s=new Square();

9) Given a string and a non-empty word string, return a string made of each char just before and just after every appearance of the word in the string. Ignore cases where there is no char before or after the word, and a char may be included twice if it is between two words.

If inputs are "abcXY123XYijk" and "XY", output should be "c13i".
If inputs are "XY123XY" and "XY", output should be "13".
If inputs are "XY1XY" and "XY", output should be "11".

10) a) Create a class called GeneralBank which acts as base class for all banks. This class has functionality getSavingInterestRate and getFixedInterestRate methods, which return the saving a/c rate of interest and fixed account rate of interest the specific bank gives. Since GeneralBank cannot say what percentage which bank would give, make it abstract.

b) Create 2 subclasses of GeneralBank called ICICIBank and KotMBank. Override the methods from base class. ICICI - Savings 4% Fixed 8.5% and KotMBank.  - Savings 6% Fixed 9%

c) Create a main method to test the above classes. Try one by one and absorb your finding.
a) ICICIBank object reference instantiated with ICICIBank class.
b) KotMBank object reference instantiated with KotMBank class.
c) GeneralBank object reference instantiated with KotMBank class.
d) GeneralBank object reference instantiated with ICICIBank class.

11) Create a package called com.automobile. Define an abstract class called Vehicle.
Vehicle class has the following abstract methods:
public String getModelName()
public String getRegistrationNumber()
public String getOwnerName()

Create TwoWheeler subpackage under Automobile package
Hero  class extends  Automobile.vehicle class
public int getSpeed()
– Returns the current speed of the vehicle.
public void radio()
– provides facility to control the radio device
Honda class extends com.automobile.vehicle class
public int getSpeed()
– Returns the current speed of the vehicle.
public void cdplayer()
– provides facility to control the cd player device which is available in the car.
Create a test class to test the methods available in all these child class.

12) A library needs to develop an online application for two types of users/roles, Adults and children. Both of these users should be able to register an account.

Any user who is less than 12 years of age will be registered as a child and they can borrow a “Kids” category book for 10 days, whereas an adult can borrow “Fiction” category books which need to be returned within 7 days.

Note: In future, more users/roles might be added to the library where similar rules will be enforced.
Develop Interfaces and classes for the categories mentioned above.
a) Create an interface LibraryUser with the following methods declared,
Method Name
registerAccount
requestBook
b) Create 2 classes “KidUsers” and “AdultUser” which implements the LibraryUser interface.
c)  Both the classes should have two instance variables as specified below.

| Instance variables | Data type |
|---|---|
| age | int |
| bookType | String |

d) The methods in the KidUser class should perform the following logic.
registerAccount function:
if age < 12, a message displaying “You have successfully registered under a Kids Account” should be displayed in the console.
If(age>12), a message displaying, “Sorry, Age must be less than 12 to register as a kid” should be displayed in the console.
requestBook function:
if bookType is “Kids”, a message displaying “Book Issued successfully, please return the book within 10 days” should be displayed in the console.

Else, a message displaying, “Oops, you are allowed to take only kids books” should be displayed in the console.

e) The methods in the AdultUser class should perform the following logic.

registerAccount function:

if age > 12, a message displaying “You have successfully registered under an Adult Account” should be displayed in the console.

If age<12, a message displaying, “Sorry, Age must be greater than 12 to register as an adult” should be displayed in the console.

requestBook function:

if bookType is “Fiction”, a message displaying “Book Issued successfully, please return the book within 7 days” should be displayed in the console.

Else, a message displaying, “Oops, you are allowed to take only adult Fiction books” should be displayed in the console.

f) Create a class “LibraryInterfaceDemo.java” with a main method which performs the below functions,

Test case #1:

Create an instance of KidUser class.

Set the age as specified in the below table and invoke the registerAccount method of the KidUser object

| Age |
|---|
| 10 |
| 18 |

Set the book Type as specified in the below table and invoke the requestBook method of the KidUser object,

| BookType |
|---|
| “Kids” |
| “Fiction” |

Test case #2:

Create an instance of AdultUser class.

Set the age as specified in the below table and invoke the registerAccount method of the AdultUser object

| Age |
|---|
| 5 |
| 23 |

Set the book Type as specified in the below table and invoke the requestBook method of the AdultUser object

| BookType |
|---|
| “Kids” |
| “Fiction” |

13) Write a program that takes as input the size of the array and the elements in the array. The program then asks the user to enter a particular index and prints the element at that index. Index starts from zero. This program may generate Array Index Out Of Bounds Exception or NumberFormatException . Use exception handling mechanisms to handle this exception.

Sample Input and Output 1:

```
Enter the number of elements in the array
2
```

```
Enter the elements in the array
50
80
Enter the index of the array element you want to access
1
The array element at index 1 = 80
The array element successfully accessed
```

Sample Input and Output 2:

```
Enter the number of elements in the array
2
Enter the elements in the array
50
80
Enter the index of the array element you want to access
9
java.lang.ArrayIndexOutOfBoundsException
```

Sample Input and Output 3:

```
Enter the number of elements in the array
2
Enter the elements in the array
30
j
java.lang.NumberFormatException
```

Exception Handling: Try-catch Use multiple catch block

14) A student portal provides user to register their profile. During registration the system needs to validate the user should be located in India. If not the system should throw an exception.
Step 1: Create a user defined exception class named “InvalidCountryException”.
Step 2: Overload the respective constructors.
Step 3: Create a main class “UserRegistration”, add the following method,
registerUser– The parameter are String username,String userCountry and add the following logic,
• if userCountry is not equal to “India” throw a InvalidCountryException with the message “User Outside India cannot be registered”
• if userCountry is equal to “India”, print the message “User registration done successfully”
Invoke the method registerUser from the main method with the data specified and see how the program behaves,
Name Country Expected Output
Mickey US InvalidCountryException should be thrown.
The message should be “User Outside India cannot be registered”
Mini India The message should be “User registration done successfully”
Sample Input and Output