**What symbol do flowcharts end with?**

**1)oval**

**2)Terminal**

**3)rectangle**

**4)diamond**

**Find the correct order  
To test if a number is palindrome or not, do the following steps:  
1. If both are the same then print palindrome number else print not a palindrome number.  
2. Get the number from a user.  
3. Compare it with the number entered by the user.  
4. Reverse it.**

**1)1, 4, 3, 2**

**2)2, 4, 3, 1**

**3)2, 3, 4, 1**

**4)4, 2, 1, 3**

**Which of the following is correct pseudocode for find an average of n numbers**

**1)**

**1. Input the n numbers into the computer**

**2. Calculate the average by multiplying the numbers and add the answer by n**

**3. Display the average**

**2)**

**1. Input the n numbers into the computer**

**2. Calculate the average by adding the numbers and dividing the sum by 1**

**3. Display the average**

**3)**

**1. Input the n numbers into the computer**

**2. Calculate the average by multiplying the numbers and add the answer by 100**

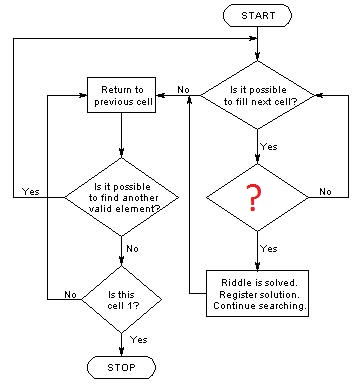
**3. Display the average**

**4)**

**1. Input the n numbers into the computer**

**2. Calculate the average by adding the numbers and dividing the sum by n**

**3. Display the average**

**Find a way for a Knight to visit every square on a board exactly once.  
  
  
which of the following is correct for above flowchart**

**1)Are all cells not filled?**

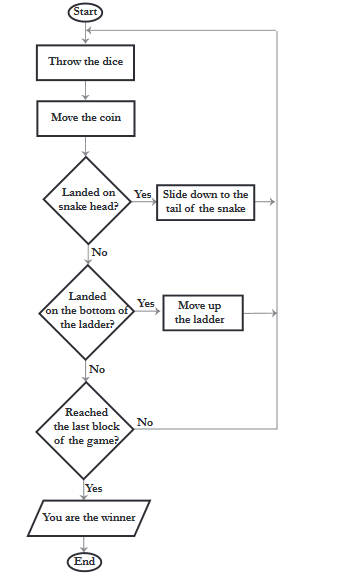
**2)Are 3 cells filled?**

**3)Are all cells filled?**

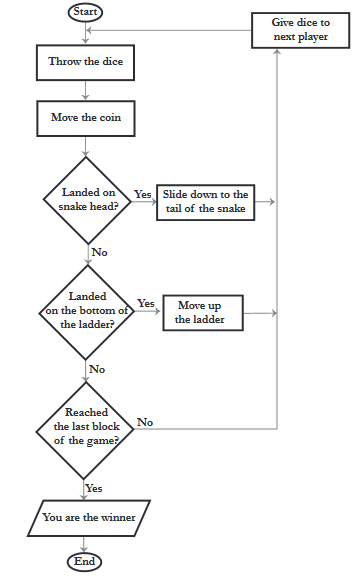
**4)Are 8 cells filled?**

**Which of the following is correct flowchart for Snakes and Ladder game**

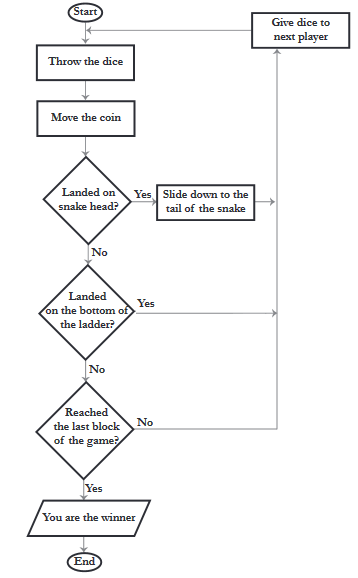
**1)**

****

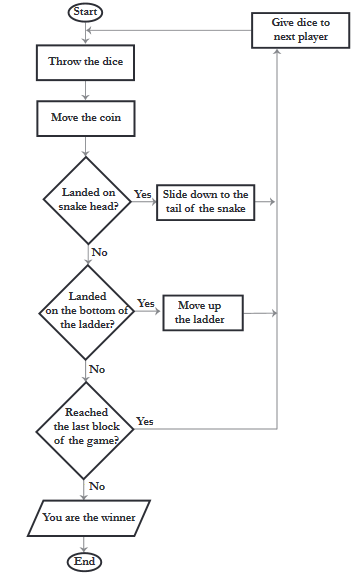
**2)**

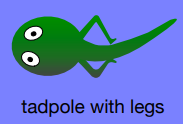
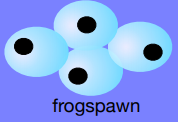
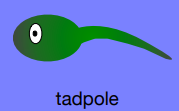
****

**3)**

****

**4)**

****

**Find the correct sequence of frog lifecycle  
  
    
  
  **

**1)**

**tadpole with legs->tadpole->froglet->frog->frogspawn.**

**2)**

**tadpole->tadpole with legs->froglet->frog->frogspawn.**

**3)**

**frogspawn->tadpole with legs->froglet->frog->tadpole.**

**4)**

**frogspawn->tadpole->tadpole with legs->froglet->frog**

**Which of the following is correct?  
Algorithms can be represented with \_\_\_\_**

**1)**

**language**

**2)**

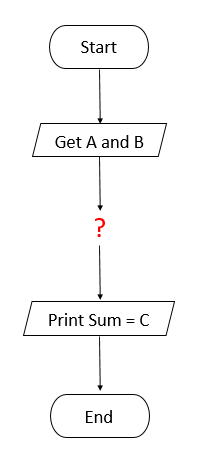
**system**

**3)**

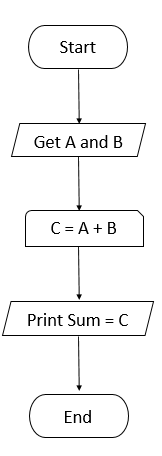
**syntax**

**4)**

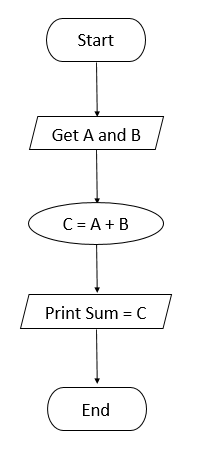
**pseudo codes**

**What is missing from the flowchart  
**

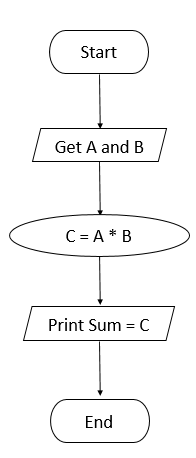
**1)**

****

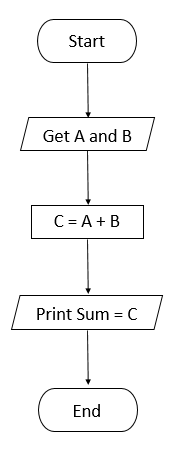
**2)**

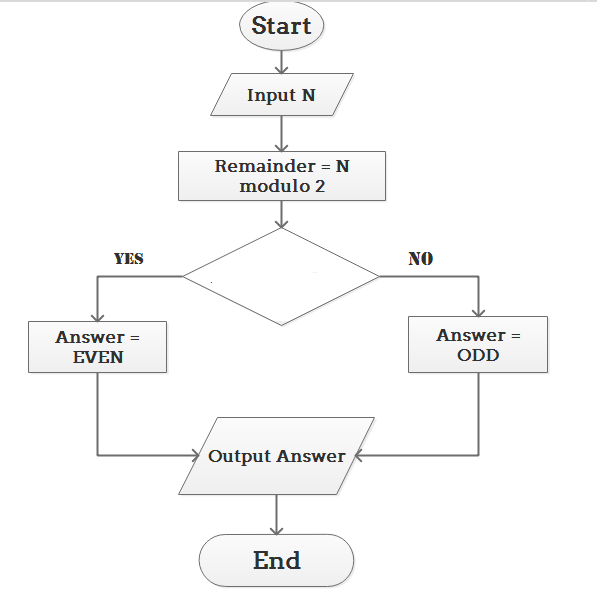
****

**3)**

****

**4)**

****

**Fill up the missing content in decision box to determine and output whether Number N is Even or Odd  
  
**

**1)**

**Is Remainder = 0**

**2)**

**Is Remainder = 1**

**3)**

**Is Remainder != 0**

**4)**

**None of the listed options**

**Find the missing line in the below Pseudocode to print the area of a rectangle.  
  
Read l  
Read b  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Write area**

**1)**

**compute area equal to l times b**

**2)**

**compute a equal to l times b**

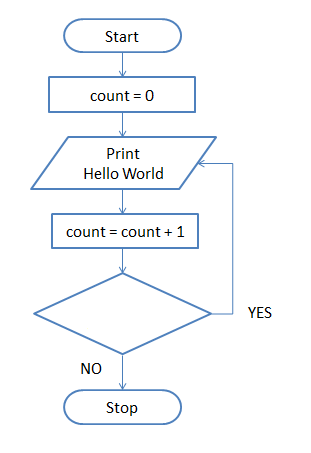
**3)**

**compute area equal to length times breadth**

**4)**

**all of the mentioned options**

|  |
| --- |
| **Fill up the blank spaces in the below Pseudocode to compute Factorial of a number : Step 1: Declare N and F as integer variable. Step 2: Initialize F=1. Step 2: Enter the value of N. Step 3: Check whether N>0, if not then F=1. Step 4: If yes then, F=F\*N Step 5: Decrease the value of N by 1 . Step 6: Repeat step \_\_\_\_ and \_\_\_\_\_ until N=0. Step 7: Now print the value of F. Type the steps in the below text box, separated by a space. (For eg: 2 3)** |
| **Correct Answer: 4 5** |

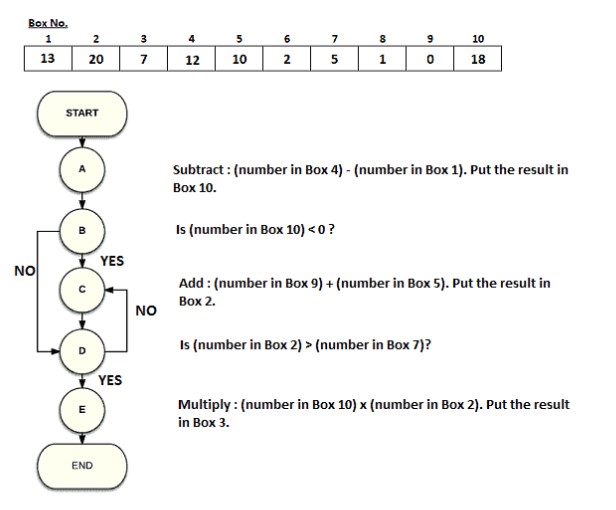
**Fill up the missing content in decision box to print Hello World 10 times.  
  
**

**1)Is count < 10**

**2)Is count <= 10**

**3)Is count < 11**

**4)None of the listed options**

**At the end of the flow chart the number placed in which of the following boxes will remain unchanged  
**

**1)Box 3**

**2)Box 10**

**3)Box 7**

**4)Box 2**

**What symbol do flowcharts use to show an action or process?**

**1)diamond**

**2)rectangle**

**3)parallelogram**

**4)o**

Write an algorithm to find the factorial of given number

**Factorial(number):**

**SET Fact = 1 and i = 1**

**WHILE i<=number**

**SET Fact=Fact\*i**

**SET i=i+1**

**ENDWHILE**

**PRINT Fact**

**END**

**Fibonacci Sequence**  
  
Write an algorithm to generate the Fibonacci Sequence upto to the given number.

**Fibonacci(number):**

**SET first = 0 , second = 1 and i = 2**

**PRINT first and second**

**WHILE (i<number)**

**SET next = first + second and PRINT next**

**SET first = second**

**SET second = next and i = i+1**

**ENDWHILE**

**END**

**Number of digits**  
  
Write an algorithm to display the number of digits in a given number.

**NumberOfDigits(number):**

**SET count=0**

**WHILE (number > 0):**

**SET count=count+1 and SET number=number/10**

**ENDWHILE**

**PRINT count**

**END**

**Maximum element in an array**  
  
Write an algorithm to display the maximum element in an array.

**ArrayMaxElement(arr, N):**

**SET i=1 and max=arr[0]**

**WHILE (i<N):**

**IF (arr[i]>max) THEN**

**SET max=arr[i]**

**ENDIF**

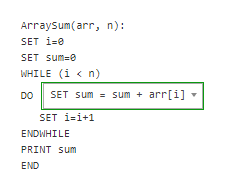
**SET i=i+1**

**ENDWHILE**

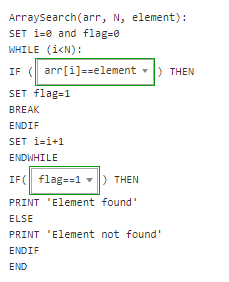
**PRINT max**

**END**

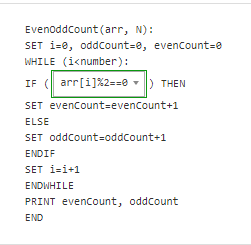
**Sum of elements in an array**  
  
Write an algorithm to find the sum of the numbers in an array.

****

**Search Element**  
  
Write an algorithm to search for an element in an array.

****

**Count of Even and Odd Elements**  
  
Write an algorithm to display the number of even and odd elements in an array.

****

SWAPING TWO NUMBERS

**SWAPPING OF TWO ROLL NUMBERS**

Rita was about to award 2 students with 1st and 2nd prize. But unfortunately, he interchanged both of them. Consider the first number as the roll number of the student who won a the1st prize and the next roll number of the student who won 2nd prize. Can you write a program to swap two numbers without using a third variable?

**Input format  :**

  Input consists of two integers.

**Output format :**

  Output consists of two integers which are swapped.

**Sample Input and Output :**

[ All text of bold corresponds to input ]

Enter values:

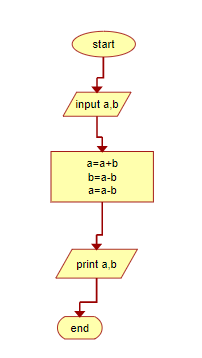
**15**

**2**

Values after swapping:

2

15

****

## EVEN OR ODD

**Even or Odd**

Write a program to check whether the given number is Even or Odd  
  
**Input Format:**  
Input consists of an single integer.  
  
**Output Format:**  
Refer to the sample input an output.  
**[All text in bold corresponds to input and the rest corresponds to output.]**

**Sample Input and Output 1:**

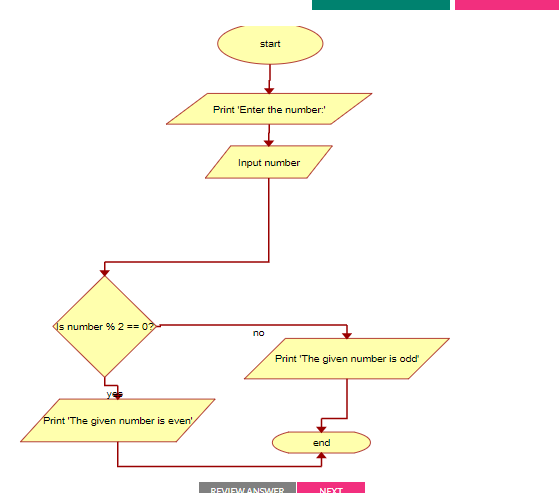
Enter the number:  
**4**

The given number is even

**Sample Input and Output 2:**

Enter the number:  
**5**

The given number is odd

****

## S1P2 - REVERSE OF A NUMBER

**Reverse of a number**

Write a program to reverse the digits of a number.  
  
**(Note : Please use initialize statement before input statement)**

**Input format :**

Input consists of an integer value.

**Output format :**

Output consists of the reverse of the given number.

[ Refer Sample Input and Output for further details ]

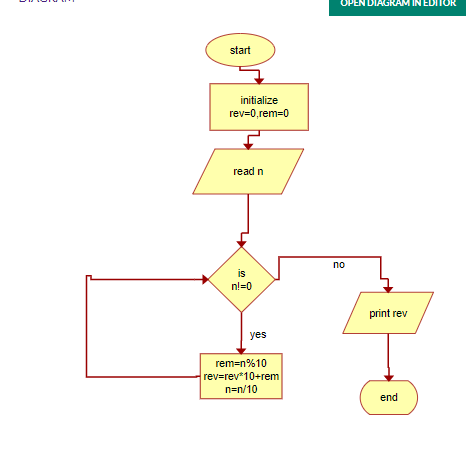
**Sample Input and Output 1 :**

**[ All text of bold corresponds to Input and the rest output]**

Enter the number :  
**5642**  
Reverse of the number is 2465

**Sample Input and Output 1 :**

Enter the number :  
**144**  
Reverse of the number is 441

****

## M8P2-P2-DIGIT COUNTING

**DIGIT COUNTING**

Write a program to find the number of digits in a given number.  
**(Note : Please use initialize statement before input statement)**

**Input Format:**

Input consists of an integer.

**Output Format:**

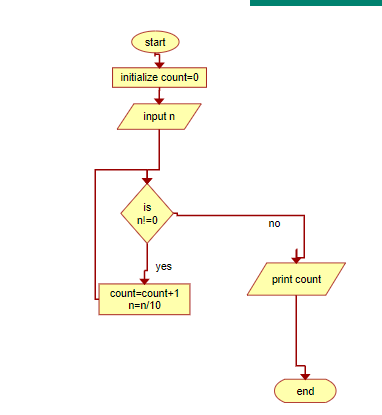
Output consists of a single line. Refer sample output for details.

**Sample Input 1:**

42

**Sample Output 1:**

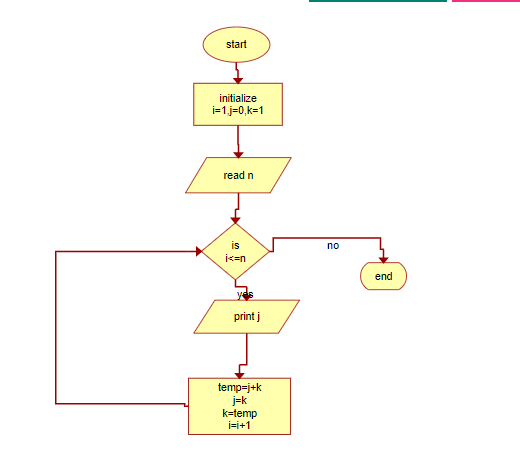
The number of digits in 42 is 2

****

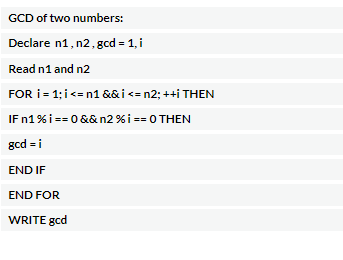
**Fibonacci Series**

Madhu and Balaji had a competition to generate a fibonacci series.  
Its a Hundred rupees bet!!!  
Why cant you help him?  
Create a variable 'n' to get the range.  
  
**(Note : Please use initialize statement before input statement)**

**Sample Input and Output 1:  
[All text in bold corresponds to input and the rest corresponds to output.]**  
Enter the range:  
**7**  
Fibonacci series:  
0  
1  
1  
2  
3  
5  
8

****

**Greatest Common Divisor**  
  
Write an algorithm to finf the Greatest Common Divisor of two numbers.

****

**How to execute a java project inside the Eclipse**

**1)**

**Run As -> Run Applet**

**2)**

**Run As -> Run on Server**

**3)**

**Run As -> Java Project**

**4)**

**Run As -> Java Application**

**What is the full form of JRE?**

**1)**

**Java Runtime Eclipse**

**2)**

**Java Run Environment**

**3)**

**Java Runtime Environment**

**4)**

**None of the mentioned options**

**1)**

**public class Main {**

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

**Write your code//**

**}**

**}**

**2)**

**public class Main {**

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

**Write your code// comments**

**}**

**}**

**3)**

**public class Main {**

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

**//Write your code**

**}**

**}**

**4)**

**public class Main {**

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

**Write your code**

**}**

**}**

**Select the appropriate code where 'Write your code' should be commented using multi line comments**

**1)**

**public class Main {**

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

**//Write**

**//your**

**//code**

**}**

**}**

**2)**

**public class Main {**

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

**/\* Write**

**your**

**code\*/**

**}**

**}**

**3)**

**public class Main {**

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

**\*/Write**

**your**

**code/\***

**}**

**}**

**4)**

**public class Main {**

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

**\*/Write**

**your**

**code\*/**

**}**

**}**

**A file named myprog.java is compiled and the compiler generates**

**1)**

**myprog.exe**

**2)**

**myprog.class**

**3)**

**myprog.obj**

**4)**

**none of the mentioned**

Which is the correct "Hello World" program?

**1)**

**Class HelloWorld**

**{**

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

**{**

**System.out.println("Hello World");**

**}**

**}**

**2)**

**class HelloWorld**

**{**

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

**{**

**System.out.println("Hello World")**

**}**

**}**

**3)**

**class HelloWorld**

**{**

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

**{**

**System.out.println("Hello World");**

**}**

**}**

**4)**

**class HelloWorld**

**{**

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

**{**

**system.out.println("Hello World");**

**}**

**}**

**The \_\_\_\_\_\_\_\_ defines where the Java compiler and Java runtime look for .class files to load**

**1)**

**class**

**2)**

**classpath**

**3)**

**libraries**

**4)run time environment**

**Can bytecode run directly on the machine?**

**1)Yes**

**2)No**

**Eclipse is an \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(IDE) used in computer programming.**

**1)Integrated Design Environment**

**2)Internal Design Environment**

**3)Internal Development Environment**

**4)Integrated Development Environment**

**Consider the java file HelloWorld.java with Class HelloWorld. How to compile the code inside the file using command.**

**1)javac HelloWorld.java**

**2)java HelloWorld.java**

**3)javac HelloWorld**

**4)java HelloWorld**

**How many types of errors are there?**

**1)3**

**2)2**

**3)1**

**4)4**

**Where does the execution starts in java program?**

**1)Main function**

**2)main function**

**3)Main class**

**4)imports**

**Java platform is associated with \_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**1)JVM and JDK**

**2)JVM and Java core libraries**

**3)JVM and JRE**

**4)JRE and Java core libraries**

**Which errors cannot be detected by a compiler nor by the JVM?**

**1)compilation errors**

**2)run time errors**

**3)logical errors**

**4)None of the mentioned options**

**State True or False  
Comment Lines can be used to prevent some code lines from being executed.**

**1)TRUE**

**2)FALSE**

**Select the feature that is not applicable for Java**

**1)Strongly-typed programming language**

**2)Interpreted and compiled language**

**3)Platform dependent**

**4)Automatic memory management**

**What is the full form of JDK?**

**1)Java Diagram Kit**

**2)Java Demonstration Kit**

**3)Java Design Kit**

**4)Java Development Kit**

**Java is case insensitive**

1)**TRUE**

**How is a single line comment created in Java?**

**1)\\**

**2)\***

**3)//**

**4)/\***

**Which error occurs when the program is running?**

**1)compilation errors**

**2)run time errors**

**3)logical errors**

**4)None of the mentioned options**

|  |
| --- |
| **The standard java compiler** |
| **Correct Answer: javac** |
|  |

**Which errors prevent the code from successful compilation, because of error in the syntax like missing a semicolon.**

**1)compilation errors**

**2)run time errors**

**3)logical errors**

**4)None of the mentioned options**

**Consider the java file HelloWorld.java with Class HelloWorld. How to run the code inside the file using command after compilation.**

**1)javac HelloWorld.java**

**2)java HelloWorld.java**

**3)javac HelloWorld**

**4)java HelloWorld**

**Java programming language is a**

**1)object oriented language**

**2)structured language**

**3)procedural language**

**4)web based language**

**Java can be easily extended since it is based on the \_\_\_\_\_\_ model.**

**1)object-oriented**

**2)procedural**

**3)secure**

**4)platform-oriented**

**JVM stands for**

**1)Java Visual Machine**

**2)Java Virtual Machine**

**3)JaVa Machine**

**4)JAVA Member**

**Which includes JRE to execute the Java program?**

**1)JDK**

**2)JVM**

**3)Library Classes**

**4)None of the mentioned options**

The \_\_\_\_\_\_\_\_ defines where the Java compiler and Java runtime look for .class files to load

**1)javac -jar jarName.jar**

**2)java -jar**

**3)java -jar jarName.jar**

**4)java jarName.jar**

**Where does the Java compiler will be available?**

**1)JVM**

**2)JRE**

**3)None of the mentioned options**

**4)JDK**

**What does java compiler do?**

**1)Converts Java code to byte code**

**2)Converts Java code to machine code**

**3)All of the mentioned options**

**4)None of the mentioned options**

#### **Advanced Password Strength**

Using regex pattern, find the strength of the password based on the rules given below.        
  
1. If the length of the password is less than 8, then the password strength is 'weak'.  
2. If the password length is greater than eight but does not contain any special characters or numbers, the strength is  considered as 'ok'.  
3. If the password length is greater than 8 and contains special characters/numbers, the strength is considered as 'good'.  
  
Use appropriate getters and setters.

Create a class **Main.java**

Create another class file **Customer.java** with following data members.

|  |  |
| --- | --- |
| **Data Type** | **Variable Name** |
| String | name |
| String | password |

**Input and Output Format:**

Refer sample input and output for formatting specifications.

**[All text in bold corresponds to input and the rest corresponds to output]**  
  
**Sample Input & Output:**  
  
Enter number of customers  
**3**  
Enter name of customer1  
**Praveen**  
Enter the password  
**praveen**  
Enter name of customer2  
**Madhan**  
Enter the password  
**Madhan12@#**  
Enter name of customer3  
**Jimesh**  
Enter the password  
**Jimeshshar**  
Customer name: Praveen, password: praveen  
Password strength is weak  
Customer name: Madhan, password: Madhan12@#  
Password strength is good  
Customer name: Jimesh, password: Jimeshshar  
Password strength is ok

**public class Customer {**

**String name;**

**String password;**

**public String getName() {**

**return name;**

**}**

**public void setName(String name) {**

**this.name = name;**

**}**

**public String getPassword() {**

**return password;**

**}**

**public void setPassword(String password) {**

**this.password = password;**

**}**

**}**

**import java.util.ArrayList;**

**import java.util.List;**

**import java.util.Scanner;**

**import java.util.regex.Matcher;**

**import java.util.regex.Pattern;**

**public class Main {**

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

**// TODO Auto-generated method stub**

**Scanner sc=new Scanner(System.in);**

**List<Customer> cus=new ArrayList<Customer>();**

**System.out.println("Enter number of customers");**

**int count=sc.nextInt();**

**Customer c=new Customer();**

**for(int i=0;i<count;i++)**

**{**

**System.out.println("Enter name of customer"+(i+1));**

**String userName=sc.next();**

**System.out.println("Enter the password");**

**String password=sc.next();**

**c.setName(userName);**

**c.setPassword(password);**

**cus.add(c);**

**c=null;**

**c=new Customer();**

**}**

**String regex1="((?=.\*[a-z]).{1,8}$)";**

**String regex2="((?=.\*[a-z]).{9,}$)";**

**String regex3="((?=.\*[@#$%^&+=])(?=.\*[a-z]).{9,}$)";**

**for(int i=0;i<cus.size();i++)**

**{**

**System.out.println("Customer name: "+cus.get(i).getName()+","+" password: "+cus.get(i).getPassword());**

**Pattern p = Pattern.compile(regex1);**

**Matcher m = p.matcher(cus.get(i).getPassword());**

**Pattern p1 = Pattern.compile(regex2);**

**Matcher m1 = p1.matcher(cus.get(i).getPassword());**

**Pattern p2 = Pattern.compile(regex3);**

**Matcher m2 = p2.matcher(cus.get(i).getPassword());**

**if(m2.matches())**

**{**

**System.out.println("Password strength is good");**

**}**

**else if(m1.matches())**

**{**

**System.out.println("Password strength is ok");**

**}**

**else if(m.matches())**

**{**

**System.out.println("Password strength is weak");**

**}**

**//if(cus.get(0).getPassword().)**

**}**

**}**

**}**

**Call History**

Practice makes a man perfect !!! Let us use FileWriter & BufferedWriter to read call log data from console and rewrite it into a CSV file. Please refer to the specification given below.  
  
**Note :**Read the input from the user and write using 'FileWriter writer = new FileWriter("call.csv");'  
  
Name the output file as **'call.csv'.**  
  
**Sample Input and Output:**  
**[All text in bold are input and the remaining are output]**

Enter the mobile number

**8972007627**

Enter the duration (in Seconds)

**5**

Do you want to add another call history ?

**yes**

Enter the mobile number

**8976543289**

Enter the duration (in Seconds)

**6**

Do you want to add another call history ?

**no**

**Output(call.csv):**

8972007627,5

8976543289,6

 import java.io.\*;

import java.util.Scanner;

public class Main {

public static void main(String [] args) {

String fileName = "call.csv";

Scanner sc= new Scanner(System.in);

try {

FileWriter fileWriter =

new FileWriter(fileName);

BufferedWriter bufferedWriter =

new BufferedWriter(fileWriter);

String cond="yes";

while(cond.equals("yes")) {

System.out.println("Enter the mobile number");

bufferedWriter.write(sc.next());

System.out.println("Enter the duration (in Seconds)");

bufferedWriter.write(","+sc.next());

bufferedWriter.newLine();

System.out.println("Do you want to add another call history ?");

cond=sc.next();

}

bufferedWriter.close();

}

catch(IOException ex) {

System.out.println(ex.getMessage());

}

}

}

**Discounts - Inheritance**

One of the easier ways to identify the scenarios that reflect inheritance is to look for a "is-a" relationship in the requirements document. On trying to check if we have such hierarchies, we find that there are different types of customers/account holders in the Bank. Customers can be Normal, Priviledged, SeniorCitizen and so on.   
The Bank also introduces an offer where privileged customers get a 30% off on the bill while senior citizens get 12% off.  
  
Lets implement the inheritance for the given scenario yet again for a better understanding.  
  
1. Create Customer, Privileged & SeniorCitizen class with data members as given below.  
2. Implement generateBillAmount Method as per the specification.

Create a class **Customer** with the following private data members

|  |  |
| --- | --- |
| **Data Type** | **Variable Name** |
| String | name |
| String | address |
| Integer | age |
| String | mobileNumber |

Methods in class **Customer**

|  |  |
| --- | --- |
| **Method Name** | **Method description** |
| displayCustomer() | To display the details of the customer. |

Use Appropriate **Getters & Setters**for **Customer**class.

Create a class **SeniorCitizenCustomer** which extends the class **Customer**.  
Methods in class **SeniorCitizenCustomer**

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Method description** | **Return Type** |
| generateBillAmount(amount) | To calculate the payment amount where the discount is 12% . | Double |

Create a class **PrivilegeCustomer** which extends the class **Customer**.  
Methods in class **PrivilegeCustomer**

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Method description** | **Return Type** |
| generateBillAmount(amount) | To calculate the payment amount where the discount is 30% . | Double |

Create a driver class named **Main** which creates an instance of the above mentioned classes.  
Use setters to set the values to objects and display all details using getters from the main method.

**Note :**

**Strictly adhere to the object oriented specifications given as part of the problem statement.**

**Use the same class names and member variable names.**

**Input and Output Format:**

Refer sample input and output for formatting specifications.

**[All text in bold corresponds to input and the rest corresponds to output.]  
Sample Input and Output 1:**

1)Privilege Customer  
2)SeniorCitizen Customer  
Enter Customer Type  
**1**  
Enter The Name  
**Ram**  
Enter The Age  
**25**  
Enter The Address  
**CBE**  
Enter The Mobile Number  
**9576531641**  
Enter The Purchased Amount  
**5000**  
Bill Details  
Name Ram  
Mobile 9576531641  
Age 25  
Address CBE  
Your bill amount is Rs 5000.0. Your bill amount is discount under privilege customer  
You have to pay Rs 3500.00

**Sample Input and Output 2:**

1)Privilege Customer  
2)SeniorCitizen Customer  
Enter Customer Type  
**3**  
Invalid Customer Type

**public class Customer {**

**private String name;**

**private String address;**

**private Integer age;**

**private String mobileNumber;**

**public Customer() {**

**// TODO Auto-generated constructor stub**

**}**

**public Customer(String name, String address, Integer age, String mobileNumber) {**

**super();**

**this.name = name;**

**this.address = address;**

**this.age = age;**

**this.mobileNumber = mobileNumber;**

**}**

**public String getName() {**

**return name;**

**}**

**public void setName(String name) {**

**this.name = name;**

**}**

**public String getAddress() {**

**return address;**

**}**

**public void setAddress(String address) {**

**this.address = address;**

**}**

**public Integer getAge() {**

**return age;**

**}**

**public void setAge(Integer age) {**

**this.age = age;**

**}**

**public String getMobileNumber() {**

**return mobileNumber;**

**}**

**public void setMobileNumber(String mobileNumber) {**

**this.mobileNumber = mobileNumber;**

**}**

**void displayCustomer() {**

**//fill your code here**

**}**

**}**

**import java.text.DecimalFormat;**

**public class PrivilegeCustomer extends Customer{**

**DecimalFormat format = new DecimalFormat("0.00");**

**private Double amount;**

**public final Double getAmount() {**

**return amount;**

**}**

**public final void setAmount(Double amount) {**

**this.amount = amount;**

**}**

**//To calculate the payment amount where the discount is 30%**

**double generateBillAmount(Double amount) {**

**amount = amount - (0.3 \* amount);**

**return amount;**

**}**

**void displayCustomer() {**

**System.out.println("Bill Details\nName "+getName()+"\nMobile "+getMobileNumber()+"\nAge "+getAge()+"\nAddress "+getAddress()+"\nYour bill amount is Rs "+getAmount()+". Your bill amount is discount under privilege customer\nYou have to pay Rs "+format.format(generateBillAmount(amount)));**

**}**

**}**

**import java.text.DecimalFormat;**

**public class SeniorCitizenCustomer extends Customer{**

**DecimalFormat format = new DecimalFormat("0.00");**

**private Double amount;**

**public final Double getAmount() {**

**return amount;**

**}**

**public final void setAmount(Double amount) {**

**this.amount = amount;**

**}**

**//To calculate the payment amount where the discount is 12%**

**double generateBillAmount(Double amount) {**

**return amount - (0.12 \* amount);**

**}**

**void displayCustomer() {**

**System.out.println("Bill Details\nName "+getName()+"\nMobile "+getMobileNumber()+"\nAge "+getAge()+"\nAddress "+getAddress()+"\nYour bill amount is Rs "+getAmount()+". Your bill amount is discount under senior citizen customer\nYou have to pay Rs "+format.format(generateBillAmount(amount)));**

**}**

**}**

**import java.text.DecimalFormat;**

**import java.util.Scanner;**

**public class Main {**

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

**DecimalFormat format = new DecimalFormat("0.00");**

**Scanner sc = new Scanner(System.in);**

**Customer customer;**

**System.out.println("1)Privilege Customer\n2)SeniorCitizen Customer\nEnter Customer Type");**

**int choice = sc.nextInt();**

**if(choice<1 || choice>2)**

**{**

**System.out.println("Invalid Customer Type");**

**System.exit(0);**

**}**

**sc.nextLine();**

**System.out.println("Enter The Name");**

**String name = sc.nextLine();**

**System.out.println("Enter The Age");**

**Integer age = sc.nextInt();**

**sc.nextLine();**

**System.out.println("Enter The Address");**

**String address = sc.nextLine();**

**System.out.println("Enter The Mobile Number");**

**String mobileNumber = sc.nextLine();**

**System.out.println("Enter The Purchased Amount");**

**Double amount = sc.nextDouble();**

**switch (choice) {**

**case 1:**

**PrivilegeCustomer privilageCust = new PrivilegeCustomer();**

**privilageCust.setName(name);**

**privilageCust.setAddress(address);**

**privilageCust.setAge(age);**

**privilageCust.setMobileNumber(mobileNumber);**

**privilageCust.setAmount(amount);**

**privilageCust.displayCustomer();**

**break;**

**case 2:**

**SeniorCitizenCustomer seniorCitiCust = new SeniorCitizenCustomer();**

**seniorCitiCust.setName(name);**

**seniorCitiCust.setAddress(address);**

**seniorCitiCust.setAge(age);**

**seniorCitiCust.setMobileNumber(mobileNumber);**

**seniorCitiCust.setAmount(amount);**

**seniorCitiCust.displayCustomer();**

**break;**

**}**

**}**

**}**

**Finally !!!**

One of the technical requirements that you might often expect is to "execute a set of lines" irrespective of the flow that goes through try / catch. A simple scenario that you would know is to close a database connection / release resources if it goes through successfully in a try  block and happens to take an exception route. Lets practice the "finally" keyword.  
  
Create a class **Transaction**with private member variables.

|  |  |
| --- | --- |
| **DataType** | **Variable Name** |
| String | accountNumber |
| Double | amount |

Include apporapiate **getter** and **setters** for above class.  
Include appropriate **default**and **parameterized constructors** for the above class.  
  
Includewith the following method in class **Transaction**

|  |  |
| --- | --- |
| **Return Type** | **method name** |
| Boolean | validate(transactionAmount) |

In **validate** method if the transaction amount is greater than current balance or if the current balance is in minimal balance (500) then throw a manual exception then display "Insufficient Balance". Otherwise return true. Use finally to display the available balance after transaction completed.

**Sample Input and Output :**

Enter the transaction details

Enter the account number

**123456**

Enter the available amount

**5000**

Enter the transaction amount

**500**

Do you want to enter more ?(yes/no)

**yes**

Enter the transaction amount

**1000**

Do you want to enter more ?(yes/no)

**yes**

Enter the transaction amount

**2000**

Do you want to enter more ?(yes/no)

**yes**

Enter the transaction amount

**800**

Do you want to enter more ?(yes/no)

**yes**

Enter the transaction amount

**850**

Insufficient Balance

Your available balance 700.0

**class Transaction**

**{**

**String accountNumber;**

**Double amount;**

**Transaction()**

**{**

**}**

**Transaction(String a,Double b)**

**{**

**accountNumber=a;**

**amount=b;**

**}**

**void setAccountNumber(String a)**

**{**

**accountNumber=a;**

**}**

**void setAmount(Double a)**

**{**

**amount=a;**

**}**

**Double getAmount()**

**{**

**return amount;**

**}**

**boolean validate(Double a) throws Exception**

**{**

**if(amount<=500 || a>amount)**

**throw new Exception ("Insufficient Balance");**

**return true;**

**}**

**}**

**import java.util.Scanner;**

**class Main**

**{**

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

**Scanner s=new Scanner(System.in);**

**System.out.println("Enter the transaction details\nEnter the account number\nEnter the available amount");**

**Transaction t=new Transaction(s.nextLine(),Double.parseDouble(s.nextLine()));**

**try**

**{**

**do**

**{**

**System.out.println("Enter the transaction amount");**

**Double d=Double.parseDouble(s.nextLine());**

**t.validate(d);**

**t.setAmount(t.getAmount()-d);**

**System.out.println("Do you want to enter more ?(yes/no)");**

**}**

**while(s.nextLine().equals("yes"));**

**}**

**catch (Exception e)**

**{**

**System.out.println("Insufficient Balance");**

**}**

**finally**

**{**

**System.out.printf("Your available balance %.1f",t.getAmount());**

**}**

**}**

**}**

**SDF Exception**

Lets' try to catch a very specific exception, i.e, SimpleDateFormat Exception, which is thrown when we read a String and try to convert it into Date. Assume we are reading Date-of-Birth of a customer and if they enter it in an incorrect format, capture it & display a message.

Write a program to get the customer account name, account type and Date-Of-Birth. Validate the Date-Of-Birth details and display the details.  
  
Create a class**Account** with following attributes

|  |  |
| --- | --- |
| **Data type** | **Variable name** |
| String | accountName |
| String | accountType |
| Date | dob |

Include a **default**and **parameterized constructor**with all the variables.

Include the following methods in the class **Account**.

|  |  |
| --- | --- |
| **Method name** | **Method description** |
| void display() | This method is used to display the details |
| boolean validateDOB(String dob) | This method is used to validate the given date of birth and return true or false based on the validation. |

**Sample Input and Output:**  
**[All text in bold corresponds to input and the rest corresponds to output]**  
Enter the Customer details  
Enter the name  
**Sastha**  
Enter account type  
**Savings**  
Enter date-of-birth  
**02/13/1993**  
Wrong Format(eg:01/01/2015)  
Enter date-of-birth  
**35/02/1993**  
Wrong Format(eg:01/01/2015)  
Enter date-of-birth  
**25/02/1993**  
  
Account Details  
Name : Sastha  
Type : Savings  
D.O.B : Feb-25-1993

**import java.util.Date;**

**import java.text.DateFormat;**

**import java.text.ParseException;**

**import java.text.SimpleDateFormat;**

**public class Account {**

**String accountName;**

**String accountType;**

**Date dob;**

**public String getAccountName() {**

**return accountName;**

**}**

**public void setAccountName(String accountName) {**

**this.accountName = accountName;**

**}**

**public String getAccountType() {**

**return accountType;**

**}**

**public void setAccountType(String accountType) {**

**this.accountType = accountType;**

**}**

**public Date getDob() {**

**return dob;**

**}**

**public void setDob(Date dob) {**

**this.dob = dob;**

**}**

**public Account() {**

**super();**

**// TODO Auto-generated constructor stub**

**}**

**public Account(String accountName, String accountType, Date dob) {**

**super();**

**this.accountName = accountName;**

**this.accountType = accountType;**

**this.dob = dob;**

**}**

**void display() throws ParseException{**

**System.out.println("\nAccount Details");**

**System.out.println("Name : " + getAccountName());**

**System.out.println("Type : " + getAccountType());**

**SimpleDateFormat outputbirthformat = new SimpleDateFormat("MMM-dd-yyyy");**

**String dob = outputbirthformat.format(getDob());**

**System.out.println("D.O.B : " + dob);**

**}**

**boolean validateDOB(String dob) {**

**boolean result = false;**

**String[] numbers = dob.split("/");**

**int day = Integer.parseInt(numbers[0]);**

**int month = Integer.parseInt(numbers[1]);**

**try {**

**int[] numberOfDaysEachMonth = new int[] {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};**

**if (month > 0 && month < 13) {**

**if (day > 0 && day < numberOfDaysEachMonth[month - 1]) {**

**result = true;**

**} else {**

**System.out.println("Wrong Format(eg:01/01/2015)");**

**result = false;**

**}**

**} else {**

**System.out.println("Wrong Format(eg:01/01/2015)");**

**result = false;**

**}**

**} catch (Exception e) {**

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

**}**

**return result;**

**}**

**}**

**import java.text.DateFormat;**

**import java.text.ParseException;**

**import java.text.SimpleDateFormat;**

**import java.util.Scanner;**

**import java.util.Date;**

**public class Main {**

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

**Scanner sc = new Scanner(System.in);**

**SimpleDateFormat inputbirthformat = new SimpleDateFormat("dd/MM/yyyy");**

**System.out.println("Enter the Customer details");**

**System.out.println("Enter the name");**

**String name = sc.next();**

**System.out.println("Enter account type");**

**String accType = sc.next();**

**Account ac = new Account();**

**boolean status = false;**

**String dob;**

**do{**

**System.out.println("Enter date-of-birth");**

**dob = sc.next();**

**status = ac.validateDOB(dob);**

**}while(status != true);**

**Date date1 = inputbirthformat.parse(dob);**

**Account pac = new Account(name,accType,date1);**

**pac.display();**

**// Date date2 = outputbirthformat.parse(dob);**

**// System.out.println("D.O.B : " + date2);**

**}**

**}**

**Interface Practical Problem 2**

All the banks will have their own rules to be followed but there are few areas where every bank has to follow the RBI rules. So when RBI changes any condition it must be followed by all other banks.  
  
Lets think that RBI comes and says that there will be a fixed credit score . So every bank must follow that credit score for all the transactions. In our problem let us consider the fixed credit score to be 10%. So we have to calculate the credit score for the customer.  
  
Create a interface **Bank**with method - **calculateCreditScore()** of return type as double.  
Create a class **RBI** which implemets the interface Banks with 3 private data member variables -**accountNumber** of type String, **creditScore** of type double , **holderName** of type String and a fixed variable **CREDIT** of type double. Include the method **calculateCreditScore()** and **display()**.  
   
Create the class **ICICI** which extends the class **RBI** .  
  
Create the class **HDFC** which extends the class **RBI** .  
  
Use Appropriate Getters Setters for above classes.  
  
Create a driver class named Main which creates an instance of the above mentioned classes. Credit score must be calculated seperately for all the classes  (value must be round to 2 decimal place).  
  
**[All text in bold corresponds to input and the rest corresponds to output.]**  
  
**Sample Input and Output 1:**  
  
Select the Bank Name  
1.ICICI  
2.HDFC  
**1**  
Enter the Holder Name  
**Madhan Kumar**  
Enter the Account Number  
**218463**  
Enter the Previous Credit Score  
**652**  
Enter the Amount to be Paid  
**500**  
Amount Paid Successfully !!!  
Hi,Madhan Kumar  
You have gained 50.00 credit score for the payment of 500.0  
Your Total Credit Score is 702.00  
  
**Sample Input and Output 2:**  
  
Select the Bank Name  
1.ICICI  
2.HDFC  
**2**  
Enter the Holder Name  
**Raina**  
Enter the Account Number  
**62354953**  
Enter the Previous Credit Score  
**6015**  
Enter the Amount to be Paid  
**15600**  
Amount Paid Successfully !!!  
Hi,Raina  
You have gained 1560.00 credit score for the payment of 15600.0  
Your Total Credit Score is 7575.00  
  
**Sample Input and Output 3:**  
  
Select the Bank Name  
1.ICICI  
2.HDFC  
**6**  
Invalid Bank type

**import java.text.DecimalFormat;public class HDFC extends RBI {**

**private String accountNumber;**

**private double creditScore;**

**private String holderName;**

**static double CREDIT;**

**public HDFC(String accountNumber, double creditScore, String holderName) {**

**super();**

**this.accountNumber = accountNumber;**

**this.creditScore = creditScore;**

**this.holderName = holderName;**

**}**

**public static double getCREDIT() {**

**return CREDIT;**

**}**

**public static void setCREDIT(double cREDIT) {**

**CREDIT = cREDIT;**

**}public String getAccountNumber() {**

**return accountNumber;**

**}**

**public void setAccountNumber(String accountNumber) {**

**this.accountNumber = accountNumber;**

**}**

**public double getCreditScore() {**

**return creditScore;**

**}**

**public void setCreditScore(double creditScore) {**

**this.creditScore = creditScore;**

**}**

**public String getHolderName() {**

**return holderName;**

**}**

**public void setHolderName(String holderName) {**

**this.holderName = holderName;**

**}**

**private static DecimalFormat df = new DecimalFormat("0.00");**

**private static DecimalFormat dg = new DecimalFormat("0.0");**

**public double calculateCreditScore(double amount) {**

**CREDIT=amount/10;**

**creditScore=creditScore+CREDIT;**

**return creditScore;**

**}**

**public void display() {**

**System.out.println("Amount Paid Successfully !!!");**

**System.out.println("Hi,"+getHolderName());**

**System.out.println("You have gained "+df.format(CREDIT)+" credit score for the payment of "+dg.format(CREDIT\*10));**

**System.out.println("Your Total Credit Score is "+df.format(getCreditScore()));**

**}**

**}**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**import java.util.Scanner;public class Main {**

**public static String accountNumber;**

**public static double creditScore;**

**public static String holderName;public static void main(String[] args) throws IOException {**

**System.out.println("Select the Bank Name\r\n" + "1.ICICI\r\n" + "2.HDFC");**

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**int i = Integer.parseInt(br.readLine());**

**switch (i) {**

**case 1:**

**System.out.println("Enter the Holder Name");**

**holderName = br.readLine();**

**System.out.println("Enter the Account Number");**

**accountNumber = br.readLine();**

**System.out.println("Enter the Previous Credit Score");**

**creditScore = Double.parseDouble(br.readLine());**

**ICICI icici = new ICICI(accountNumber, creditScore, holderName);**

**System.out.println("Enter the Amount to be Paid");**

**icici.calculateCreditScore(Double.parseDouble(br.readLine()));**

**icici.display();**

**break;**

**case 2:**

**System.out.println("Enter the Holder Name");**

**holderName = br.readLine();**

**System.out.println("Enter the Account Number");**

**accountNumber = br.readLine();**

**System.out.println("Enter the Previous Credit Score");**

**creditScore = Double.parseDouble(br.readLine());**

**HDFC hdfc = new HDFC(accountNumber, creditScore, holderName);**

**System.out.println("Enter the Amount to be Paid");**

**hdfc.calculateCreditScore(Double.parseDouble(br.readLine()));**

**hdfc.display();**

**break;**

**default:**

**System.out.println("Invalid Bank type");**

**break;**

**}}}**

**public interface Bank {**

**abstract double calculateCreditScore();**

**}**

**public class RBI implements Bank{**

**private String accountNumber;**

**private double creditScore;**

**private String holderName;**

**static double CREDIT;**

**public double calculateCreditScore() {**

**return creditScore;**

**}**

**public void display() {**

**//fill your code here**

**}**

**}**

**import java.text.DecimalFormat;public class ICICI extends RBI {**

**private String accountNumber;**

**private double creditScore;**

**private String holderName;**

**static double CREDIT;**

**public ICICI(String accountNumber, double creditScore, String holderName) {**

**super();**

**this.accountNumber = accountNumber;**

**this.creditScore = creditScore;**

**this.holderName = holderName;**

**}**

**public static double getCREDIT() {**

**return CREDIT;**

**}**

**public static void setCREDIT(double cREDIT) {**

**CREDIT = cREDIT;**

**}public String getAccountNumber() {**

**return accountNumber;**

**}**

**public void setAccountNumber(String accountNumber) {**

**this.accountNumber = accountNumber;**

**}**

**public double getCreditScore() {**

**return creditScore;**

**}**

**public void setCreditScore(double creditScore) {**

**this.creditScore = creditScore;**

**}**

**public String getHolderName() {**

**return holderName;**

**}**

**public void setHolderName(String holderName) {**

**this.holderName = holderName;**

**}**

**private static DecimalFormat df = new DecimalFormat("0.00");**

**private static DecimalFormat dg = new DecimalFormat("0.0");**

**public double calculateCreditScore(double amount) {**

**CREDIT=amount/10;**

**creditScore=creditScore+CREDIT;**

**return creditScore;**

**}**

**public void display() {**

**System.out.println("Amount Paid Successfully !!!");**

**System.out.println("Hi,"+getHolderName());**

**System.out.println("You have gained "+df.format(CREDIT)+" credit score for the payment of "+dg.format(CREDIT\*10));**

**System.out.println("Your Total Credit Score is "+df.format(getCreditScore()));**

**}**

**}**

- Packages/Packages / Assess

**If we don't want to override the existing attribute values, then which non-access modifier will be used for an attribute.**

1)

**const**

2)

**final**

3)

**finalize**

**Error output streams are provided by which of the class**

1)

**util**

2)

**lang**

3)

**net**

**If we use any package name for the project, then the package statement should be present in the \_\_\_\_\_\_\_\_line of source code file.**

1)

**last**

2)

**any**

3)

**first**

**Select the non-access modifiers**

1)

**private**

2)

**final**

3)

**finalize**

**public class Student{  
public int data = 50;  
public void msg(){  
System.out.println("Hello");  
}  
}  
  
package mypack;  
import pack.\*;  
  
class B{  
public static void main(String args[]){  
Student s = new Student();  
System.out.println(s.data);  
s.msg();  
}  
}**

**1)**

**Compilation errors**

**2)**

**50**

**Hello**

**3)**

**Nothing will be displayed**

**4)**

**Hello**

**How many public class per source code file.**

**1)**

**1**

**2)**

**2**

**3)**

**3**

**4)**

**any number**

**In the command "javac -d . Sample.java", what does the 'dot' denotes.**

**1)**

**current folder**

**2)**

**the folder one level up**

**3)**

**the folder that two levels up**

**4)**

**the folder outside the project**

**Select the non-access modifiers which cannot be applicable for attributes and methods.**

**1)**

**abstract**

**2)**

**final**

**3)**

**finalize**

**4)**

**static**

**The statement which is true about protected modifiers**

**1)**

**Accessible only within the class.**

**2)**

**Accessible within the package.**

**3)**

**Accessible within the package and outside the package through child classes.**

**4)**

**Accessible throughout the project**

**The statement which is true about default modifiers**

**1)**

**Accessible only within the class.**

**2)**

**Accessible within the package.**

**3)**

**Accessible within the package and outside the package through child classes.**

**4)**

**Accessible throughout the project**

**Consider that we have a class named Stall inside the package 'myPack'. How to add that class inside Main class, so that the Stall class can be accessible inside Main.**

**1)**

**import myPack.\*;**

**2)**

**import myPack;**

**3)**

**import myPack.Stall;**

**4)**

**imports myPack.Stall**

**The method in System class that terminates the current Java Virtual Machine running on system.**

**1)**

**terminate( )**

**2)**

**exit()**

**3)**

**Exit()**

**4)**

**Terminate()**