**HTML 5 , CSS 3**

1. **Simple CALCULATOR**

Create the following Web page with the heading 'Calculator' and provided images.



The web page should contain following input elements and apply the specified Constraints:

|  |  |  |
| --- | --- | --- |
| Label Name | Element Name | Constraints |
| Input1 | input1 | This element is to get the first input. Type should be 'number' |
| Input2 | input2 | This element is to get the second input. Type should be 'number' |
| Select Operation | operation | A drop down list contains the following values: **Select..** , **ADD, SUBTRACT, MULTIPLY and DIVIDE.**  **Set these values as its option tag text and option tag's 'value' attributes** |
|  | submit | An image tag with the source as calc.jpg should be displayed |
|  | reset | An image tag with the source as reset.jpg should be displayed |

Consider the images are in the current folder and

* Use “calculator.jpg” as a header image.

**Apply following styles to the attributes**: Do not use CSS .

1. The heading should be done using the font color as 'blue' and with font size as '20'. (Use <font> tag)
2. The height and width of the images with the name “submit” and “reset” should be '80'.
3. The height and width of the calculator image should be '300' and '400'.

**Solution:**

1 <!DOCTYPE html>

2 <head>

3 <title> Simple Calculator </title>

4 </head>

5 <body>

6 <font size="20" color="blue"> Calculator</font>

7 <h1> <img src="calculator.jpg"> </h1>

8 <form name="calculator" >

9 <p>Input1 &emsp; &emsp; <input type="number" name="input1"/> </p>

10 <p>Input2 &emsp; &emsp; <input type="number" name="input2"/> </p>

11 <p> Select Operation &emsp; &emsp;

12 <select name="operation">

13 <option value="Select.."> Select.. </option>

14 <option value="ADD"> ADD </option>

15 <option value="SUBTRACT"> SUBTRACT</option>

16 <option value="MULTIPLY"> MULTIPLY </option>

17 <option value="DIVIDE"> DIVIDE </option>

18 </select></p>

19 <br>

20 <input type="image" alt="submit" img src="calc.jpg " /> &emsp; &emsp;

21 <input type="image" alt="reset" img src="reset.jpg " />

22 </form>

23 </body>

#### **calc.jpg**

****

#### **calculator.jpg**

****

#### **reset.jpg**

****

## 

## JAVA SCRIPT

## ACTB Connection Portal

ACTB Limited offers prepaid and postpaid connections for their customers. They have planned for various schemes to be offered for the customers opting for these connections.

 Create a webpage “Connection.html” that has the following fields.

**Note: Every tag has been given a mark. Make sure the element names are correct**

|  |  |  |  |
| --- | --- | --- | --- |
| **SNO** | **LABEL NAME** | **COMPONENT NAME** | **DESCRIPTION** |
| 1 | Customer Name | custname | To enter the name of the customer. The text “Enter the customer name” should be displayed by default in the text box. While entering the name, this text should disappear |
| 2 | Customer Email | email | To enter the email-id of the customer. The text “abc@gmail.com" should be displayed by default in the text box. While entering the e-mail, this text should disappear. This field should accept a valid email. |
| 3 | Telephone/ Mobile Number | mobile | To enter the mobile number. The mobile number should accept only digits. The text “Enter the Mobile Number" should be displayed by default in the text box. It should contain 10 digits and start with 9/8/7. |
| 4 | Connection Type | conntype | Select the connection type using drop down with  id attribute as “Pre” and “Post”. The attribute id needs to be "connectiontype" |
| 5 | Scheme Name | scheme | The scheme options are displayed in radio button and the schemes are Local, STD and FullTalkTime. Have the scheme id attributes as scheme1, scheme2 and scheme3 respectively. |
| 6 | Connection Duration | duration | The connection duration can be taken for any number of months, Assume min value is 1 and max value is 12. |
| 7 | Connection Bill Amount | submit | On clicking this button, the connection  bill amount should be calculated. |
| 8 | Reset | clear | On clicking this button, all fields should be reset. |

**RULES/CONSTRAINTS : All validations should be based on HTML 5 (Do Not use Java Script)**

. The component name should be same as given above.

·  All fields are mandatory.

·  Customer name should contain only alphabets and space.

·  Mobile Number should be of 10 digits and should start with 9/8/7.

. The Connection Type needs to be Pre or Post.

.  The scheme names are Local, STD and FullTalkTime.

. The Connection Duration can be specified ranging from 1 to 12 months.

Refer the below table for the scheme costs

**Scheme Chart Table**

|  |  |
| --- | --- |
| **Scheme** | **Cost per month** |
| Local | 200 |
| STD | 350 |
| FullTalkTime | 500 |

**Use JavaScript for doing the below calculation:**

 On clicking the Connection Bill Amount button, the monthly rental cost has to be displayed based on the Connection Type as in the **Rental Cost Table  given below.** The available options for connection type are Pre and Post. Refer the table below for monthly rental cost for the connection.

**Rental Cost Table**

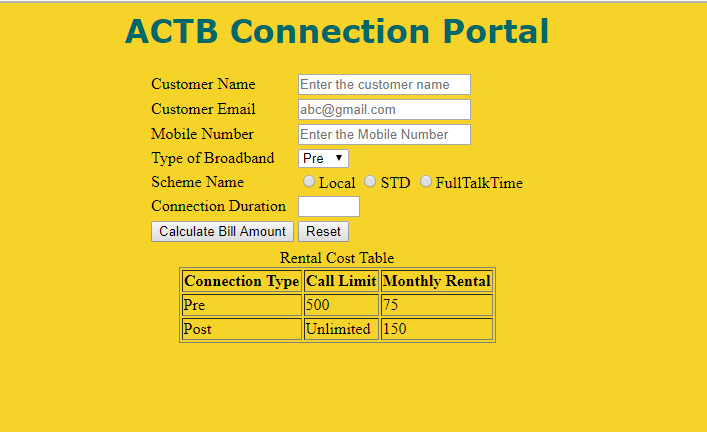
|  |  |  |
| --- | --- | --- |
| **Connection Type** | **Call Limit** | **Monthly Rental** |
| Pre | 500 | 75 |
| Post | Unlimited | 150 |

The bill amount has to be calculated and displayed as **“The Total Monthly Rental Cost is Rs. ”+ Total Monthly Rental Cost,**in a div tag. **The div tag name should be “result”.**

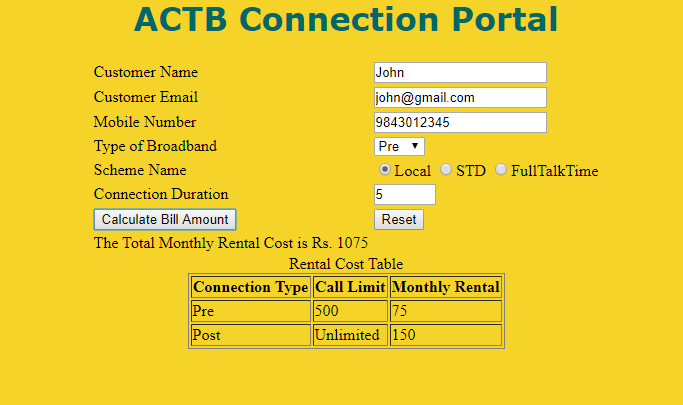
Use the following formula to calculate the Total Monthly Rental Cost:

**Total Monthly Rental Cost =  ( duration \* Cost per month ) + Monthly Rental**

**Sample Webpage:**



**Output page:**



**Styles to be applied: (Do not use Inline CSS)**

1. Body color should be #EED426.

2. The heading **ACTB Connection Portal**should be done using <h1> tag and the text color should be#006666. Style should be verdana and it should be aligned to centre of the webpage.

3. For the “Rental Cost table” the border-style should be solid and border size should be 1. It should be displayed on the right side (use float) of the webpage. The text inside the table should be aligned to centre. The table should have the id attribute with the value as “rentalcost”.

Solution:

1 <!DOCTYPE html>

2

3 <style>

4 body{

5 line-height: 190%;

6 background-color:#EED426;

7 }

8 #ipnames1{

9 position: absolute;

10 float: left;

11 left: 30%;

12 }

13 #input{

14 position: absolute;

15 float: right;

16 right: 30%;

17 }

18 h1{

19 color: #006666;

20 font-family:verdana;

21 text-align: center;

22 }

23 #rentalcost{

24 border-style: solid black 1px;

25 margin-top: 25%;

26 margin-bottom: 25%;

27 margin-left: auto;

28 margin-right: auto;

29 text-align: center;

30 }

31 </style>

32

33 <script>

34 function calc()

35 {

36 var scheme\_type=document.getElementsByName("scheme");

37 var duration\_time=document.getElementsByName("duration")[0].value;

38 var connection\_type=document.getElementById("connectiontype");

39 var monthly\_rental;

40 var cost\_per\_month;

41 for(i=0;i<scheme\_type.length;i++)

42 {

43 if(scheme\_type[i].checked)

44 {

45 if(i==0){cost\_per\_month=200;}

46 else if(i==1){cost\_per\_month=350;}

47 else if(i==2){cost\_per\_month=500;}

48 }

49 }

50 i=connection\_type.selectedIndex;

51 if(connection\_type.options[i].value=="Pre"){monthly\_rental=75;}

52 else if(connection\_type.options[i].value=="Post"){monthly\_rental=150;}

53

54 var Total\_monthly\_rental\_cost=((duration\_time\*cost\_per\_month)+monthly\_rental).toString();

55

56 document.getElementById("result").innerHTML="The Total Monthly Rental Cost is Rs. "+Total\_monthly\_rental\_cost;

57 }

58 </script>

59 <body>

60 <h1>ACTB Connection Portal</h1>

61 <div id="ipnames1">

62 Customer Name<br>

63 Customer Email<br>

64 Mobile Number<br>

65 Type of Broadband<br>

66 Scheme Name<br>

67 Connection Duration<br>

68 <input type=submit name="submit" value="Calculate Bill Amount" onclick="calc()">

69 <div name="result" id="result"></div>

70

71 </div>

72 <div id="input">

73 <input type=text placeholder="Enter the customer name" name="custname" id="custname" required><br>

74 <input type=Email placeholder="abc@gmail.com" name="email" id="email" required><br>

75 <input type=text placeholder="Enter the Mobile Number" name="mobile" id="mobile" required pattern="[7-9]{1}[0-9]{9}"><br>

76 <select name="conntype" id=connectiontype required>

77 <option id="pre" value="Pre">Pre</option>

78 <option id="post" value="Post">Post</option>

79 </select><br>

80 <input type=radio name="scheme" id=scheme1 value="Local">Local

81 <input type=radio name="scheme" id=scheme2 value="STD">STD

82 <input type=radio name="scheme" id=scheme3 value="FullTalkTime">FullTalkTime <br>

83 <input type="number" name="duration" min="1" max="12" required /> <br>

84 <input type= reset name="reset" value="Reset">

85 </div>

86

87 <table id="rentalcost" border="1" text-align="center">

88 <caption> Rental Cost Table </caption>

89 <tr>

90 <th>Connection Type</th>

91 <th>Call Limit</th>

92 <th>Monthly Rental</th>

93 </tr>

94 <tr>

95 <td>Pre</td>

96 <td>500</td>

97 <td>75</td>

98 </tr>

99 <tr>

100 <td>Post</td>

101 <td>Unlimited</td>

102 <td>150</td>

103 </tr>

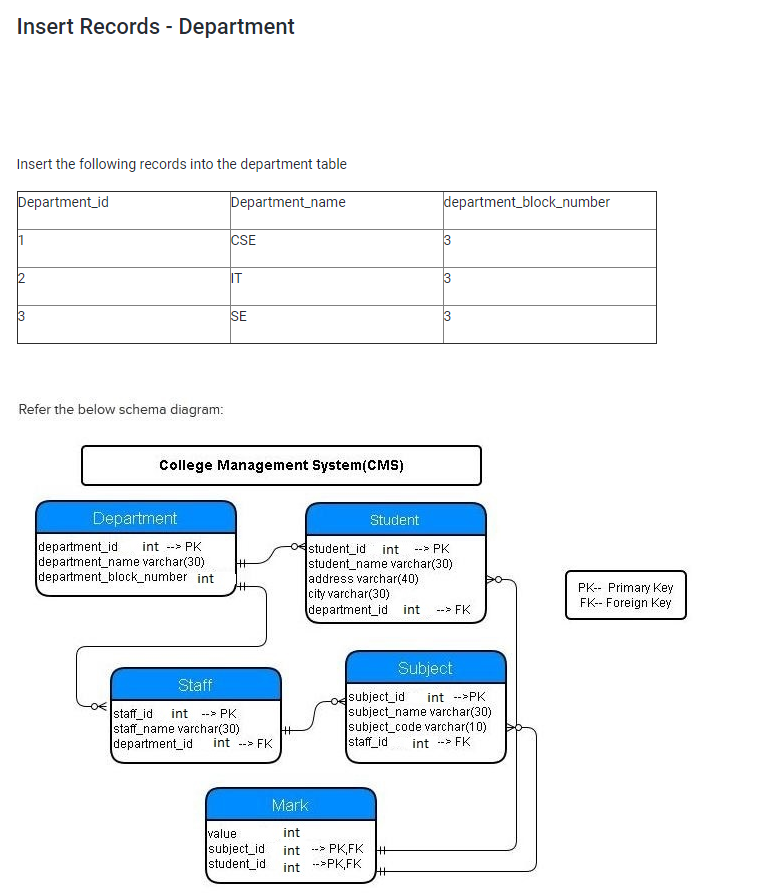
104 </table>

105

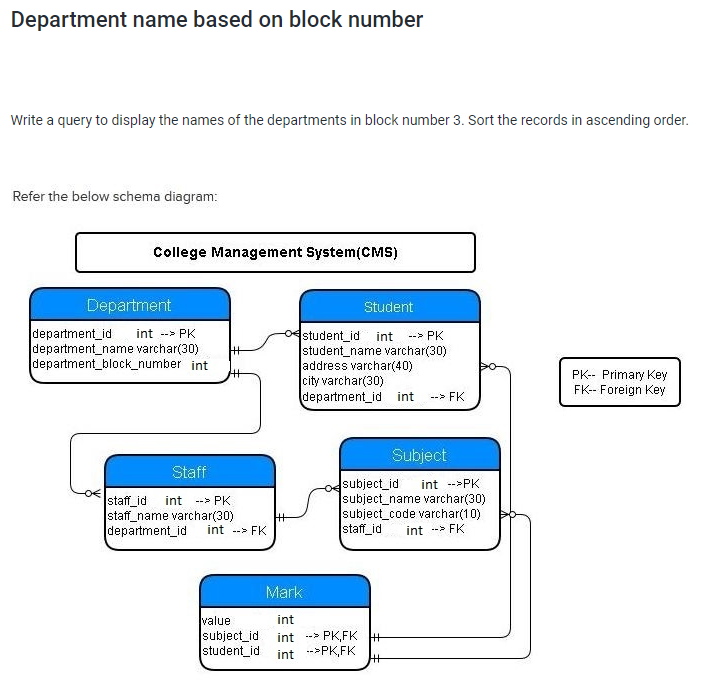
106 </body>

107 </html>

**Database Management**



Solution: *insert* *into* department (department\_id,department\_*name*,department\_block\_*number*) *values*(1,'CSE',3),(2,'IT',3),(3,'SE',3);



Solution:

*select* department\_*name* *from* department *where* department\_block\_*number* =3 *ORDER* *BY* department\_*name* *ASC*;

## 

## Solution:

*SELECT* s.student\_*name*, d.department\_*name* *FROM* department d *INNER* *JOIN* Student s *ON* (d.department\_id=s.department\_id) *where* s.city="Coimbatore" *ORDER* *BY* s.student\_*name*;

## Concatenating details

## 

## Solution:

*select* concat(address,',',city) *as* address *from* student *ORDER* *BY* address *DESC*;

## JAVA PROGRAMMING

## Fuel Consumption Calculator

Write a program to calculate the fuel consumption of your vehicle.

The program should ask the user to enter the quantity of petrol to fill up the tank and the distance covered till the tank goes dry.

Calculate the fuel consumption and display it in the format (liters per 100 kilometers).

Convert the same result to the U.S. style of miles per gallon and display the result. If the quantity or distance is zero or negative display "<respective\_input> is an Invalid Input".

[Note: The US approach of fuel consumption calculation (distance / fuel) is the inverse of the European approach (fuel / distance ). Also note that 1 kilometer is 0.6214 miles, and 1 liter is 0.2642 gallons.]

The result should be with two decimal place.

To get two decimal place refer the below-mentioned print statement :

float cost=670.23;

System.out.printf("You need a sum of Rs.%.2f to cover the trip",cost);

**Sample Input 1:**

Enter the no of liters to fill the tank

20

Enter the distance covered

150

**Sample Output 1:**

Liters/100KM

13.33

Miles/gallons

17.64

**Explanation:**

For 150 KM fuel consumption is 20 liters,

Then for 100 KM fuel consumption would be (20/150)\*100=13.33,

Distance is given in KM, we have to convert it to miles (150\*0.6214)=93.21,

Fuel consumption is given in liters, we have to convert it to gallons (20\*0.2642)=5.284,

Then find (miles/gallons)=(93.21/5.284)=17.64

**Sample Input 2:**

Enter the no of liters to fill the tank

-5

**Sample Output 2:**

-5 is an Invalid Input  
  
**Sample Input 3:**

Enter the no of liters to fill the tank

25

Enter the distance covered

-21

**Sample Output 3:**

-21 is an Invalid Input

## Solution:

#### **Main.java**

1 *import* java.util.\*;

2 *import* java.io.\*;

3

4

5 *public* *class* Main {

6

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

8 *double* l,d,s,m,n,g;

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

10 System.out.println("Enter the no of liters to fill the tank ");

11 l=sc.nextDouble();

12

13 *if*(l<=0){

14 System.out.println(""+(*int*)l+" is an Invalid Input");

15 }

16 *else* {

17 System.out.println("Enter the distance covered");

18 d=sc.nextDouble();

19 *if*(d<=0)

20 {

21 System.out.println(""+(*int*)d+" is an Invalid Input");

22 }

23 *else*{

24 s=(l/d)\*100;

25 m=d\*0.6214;

26 n=l\*0.2642;

27 g=m/n;

28 System.out.println("Liters/100KM ");

29 System.out.printf("%.2f\n",s);

30 System.out.println("Miles/gallons ");

31 System.out.printf("%.2f\n",g);

32 }

33 }

34 }

35 }

36 /\*

37 if(l>0){

38 if(k>0){

39

40 System.out.println("Liters/100KM");

41 System.out.printf("%.2f", ((l/k)\*100));

42 System.out.println("");

43 System.out.println("Miles/gallons");

44 System.out.printf("%.2f",((k\*0.6214)/(l\*0.2642)));

45 }

46 else if(k<=0)

47 {

48 int b=(int)l;

49 System.out.printf(b+ " is an Invalid Input");

50 }

51 }

52 else {

53 System.out.printf(a+ " is an Invalid Input");

54 }

55 }

56 }

57

58 /\* public static void main(String[] args)

59 {

60 float l,k;

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

62 Main Fun= new Main();

63 System.out.println("Enter the no of liters to fill the tank ");

64 l=sc.nextInt();

65 System.out.println("Enter the distance covered " );

66 k=sc.nextInt();

67 Fun.fuelConsumption(l,k);

68 }\*/

69

70

## Least offer

Maya buys “N” no of products from a shop. The shop offers a different percentage of discount on each item. She wants to know the item that has the minimum discount offer, so that she can avoid buying that and save money.

[Input Format: The first input refers to the no of items; the second input is the item name, price and discount percentage separated by comma(,)]

Assume the minimum discount offer is in the form of Integer.

**Note:**There can be more than one product with a minimum discount.

**Sample Input 1:**

4

mobile,10000,20

shoe,5000,10

watch,6000,15

laptop,35000,5

**Sample Output 1:**

shoe

**Explanation**: the discount on the mobile is 2000, the discount on the shoe is 500, the discount on the watch is 900 and the discount on the laptop is 1750. So the discount on the shoe is the minimum.

**Sample Input 2:**

4

Mobile,5000,10

shoe,5000,10

WATCH,5000,10

Laptop,5000,10

**Sample Output 2:**

Mobile

shoe

WATCH

Laptop

## Solution:

#### **Main.java**

1 *import* java.util.Scanner;

2 *public* *class* Main {

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

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

5 String a;

6 *int* n=sc.nextInt();

7 *int*[] a2=*new* *int*[n+1];

8 *int*[] a3=*new* *int*[n+1];

9 *int* i;

10 String[] a1=*new* String[n+1];

11 *int*[] d= *new* *int*[n+1];

12 *for*(i=0;i<=n;i++)

13 {

14 a=sc.nextLine();

15 String[] asplit = a.split(",");

16 *int* k=0;

17 *for*(String w:asplit)

18 {

19 *if*(k==0)

20 {

21 a1[i]=w;

22 }

23 *if*(k==1)

24 {

25 a2[i]=Integer.parseInt(w);

26 }

27 *if*(k==2)

28 {

29 a3[i]=Integer.parseInt(w);

30 }

31 k++;

32 }

33 }

34 *for*(i=1;i<=n;i++)

35 {

36 d[i]=a2[i]\*a3[i]/100;

37 }

38 *int* p=d[1];

39 *for*(i=1;i<=n;i++)

40 {

41 *if*(d[i]<=p)

42 {

43 p=d[i];

44 }

45 }

46 *for*(i=1;i<=n;i++)

47 {

48 *if*(d[i]==p)

49 {

50 System.out.println(a1[i]);

51 }

52 }

53 }

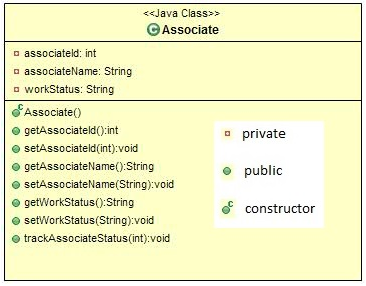
54 }

## APPLYING OBJECT ORIENTED CONCEPTS

## DreamTek Company

DreamTek Company provides an initial training for all its employees, once they join the company. During the training phase they call the employees as “Associate”. The initial training is conducted for 60 days for each Associate. In these 60 days they learn various technologies. The first 20 days they learn “Core skills”, the next 20 days they learn “Advanced modules” and the final 20 days they go to the “Project phase”.  Help the DreamTek Company to find in which phase the associates are in.

Consider the below class:



In the **Associate**class include the given attributes and methods with the access specifiers as specified in the class diagram.

The setter methods are used to set the value and the getter methods are used to retrieve the value.

The trackAssociateStatus method takes the number of days as argument and sets the work status of the associate based on the number of days. If the number of days is greater than 60 days then set the work status as “Deployed in project”.

 In the **Main** class, create an object for the Associate class; Get the details as shown in the sample input and assign the value for its attributes using the setters. Invoke the trackAssociateStatus method and find the work status and display the details as shown in the sample output.

**Sample Input1:**

Enter the associate id:

123

Enter the associate name:

john

Enter the number of days:

45

**Sample Output1:**

The associate john work status:Project phase

**Sample Input2:**

Enter the associate id:

124

Enter the associate name:

ram

Enter the number of days:

70

**Sample Output2:**

The associate ram work status: Deployed in project

## Solution:

#### **Associate.java**

1

2 *public* *class* Associate {

3 *private* *int* associateId;

4 *private* String associateName;

5 *private* String workStatus;

6 *private* *int* noOfDays;

7

8 *public* *int* getAssociateId()

9 {

10 *return* associateId;

11 }

12

13 *public* *void* setAssociateId(*int* associateId)

14 {

15 *this*.associateId = associateId;

16 }

17

18 *public* String getAssociateName()

19 {

20 *return* associateName;

21 }

22

23 *public* *void* setAssociateName(String associateName)

24 {

25 *this*.associateName = associateName;

26 }

27

28 *public* String getWorkStatus()

29 {

30 *return* workStatus;

31 }

32

33 *public* *void* setWorkStatus(String workStatus)

34 {

35 *this*.workStatus = workStatus;

36 }

37

38 *public* *int* getNoOfDays()

39 {

40 *return* noOfDays;

41 }

42

43 *public* *void* setNoOfDays(*int* noOfDays)

44 {

45 *this*.noOfDays = noOfDays;

46 }

47

48 *public* *void* trackAssociateStatus(*int* noOfDays)

49 {

50 *if*(noOfDays>0 && noOfDays<=20)

51 setWorkStatus("Core skills");

52 *else* *if*(noOfDays >20 && noOfDays <=40)

53 setWorkStatus("Advanced modules");

54 *else* *if*(noOfDays>40 && noOfDays<=60)

55 setWorkStatus("Project phase");

56 *else* *if*(noOfDays>60)

57 setWorkStatus("Deployed in project");

58 }

59

60 }

61

62

#### **Main.java**

1 *import* java.util.Scanner;

2 *public* *class* Main {

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

4 {

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

6 Associate a = *new* Associate();

7 System.out.println("Enter the associate id:");

8 *int* id=sc.nextInt();

9 a.setAssociateId(id);

10 System.out.println("Enter the associate name:");

11 sc.nextLine();

12 String name=sc.nextLine();

13 a.setAssociateName(name);

14 System.out.println("Enter the number of days:");

15 *int* noOfDays=sc.nextInt();

16 a.setNoOfDays(noOfDays);

17 a.trackAssociateStatus(a.getNoOfDays());

18 System.out.println("The associate " +a.getAssociateName()+ " work status:" +a.getWorkStatus());

19

20

21 }

22

23 }

24

25

26

## Shape - Area Volume Calculator

Create an **abstract** **public class Shape** with the below public abstract methods :

     public double area()

     public double volume()

Create a **public class Rectangle**with private attributes :

       double length

      double width

Write the public getters and setters for these attributes.

**Rectangle class should inherit the Shape class**

      Area of rectangle is **length \* width**

Create a **public class Triangle** with private attributes :

      double base

      double height

Write the public getters and setters for these attributes.

**Triangle class should inherit the Shape class**

      Area of triangle is  **1/2 \* base \* height**

 Create a **public class Cube** with private attributes :

      double length

     double width

     double height

 Write the public getters and setters for these attributes.

**Cube class should inherit the Shape class**

      Area of cube is  **2 \* length \* width + 2 \* length \* height + 2\* width \* height**

      Volume of cube is  **length \* width \* height**

 Create a **public class Sphere** with private attributes :

      double radius

 Write the public getters and setters for these attributes.

**Sphere class should inherit the Shape class**

        Area of  Sphere is  **4  \* PI \* radius2**

        Volume of Sphere is   **( 4  \* PI \* radius3) / 3**

**Note : You should use Math.PI for the value of PI**

For Rectangle and Triangle class the method volume should **return -1**.

Create an **interface Spatial** which is a marker interface. Classes that has proper implementation for volume should implement this interface.

 Create a public Main class which has the main method.

 Create an **array of Shape of size 5**.

 Get the Shape type and the corresponding attributes and store those objects in the array.

  Print the area and volume of the objects created.

  Volume of the array object should be printed **only if it is of Spatial Type**.

**Sample Input :**

**Triangle**

**10**

**20**

**Sphere**

**14**

**Rectangle**

**14**

**15**

**Cube**

**5**

**7**

**9**

**Triangle**

**18**

**24**

**Sample Output**

Area 100.0

Area 2463.0086404143976

Volume 11494.040321933855

Area 210.0

Area 286.0

Volume 315.0

Area 216.0

**Note : Volume is displayed only when the shape is cube or sphere**

## Solution:

#### **Sphere.java**

1

2 *public* *class* Sphere *extends* Shape *implements* Spatial{

3

4 *private* *double* radius;

5

6 *public* *double* getRadius() {

7 *return* radius;

8 }

9

10 *public* *void* setRadius(*double* radius) {

11 *this*.radius = radius;

12 }

13

14 @Override

15 *public* *double* area() {

16 *double* a=4\*Math.PI\*radius\*radius;

17 *return* a;

18 }

19

20 @Override

21 *public* *double* volume() {

22 *double* v=(4\*Math.PI\*radius\*radius\*radius)/3;

23 *return* v;

24 }

25

26 }

27

#### **Triangle.java**

1 *public* *class* Triangle *extends* Shape{

2

3 *private* *double* base;

4 *private* *double* height;

5

6 *public* *double* getBase() {

7 *return* base;

8 }

9 *public* *void* setBase(*double* base) {

10 *this*.base = base;

11 }

12 *public* *double* getHeight() {

13 *return* height;

14 }

15 *public* *void* setHeight(*double* height) {

16 *this*.height = height;

17 }

18 @Override

19 *public* *double* area() {

20 *double* a=(0.5\*base\*height);

21 *return* a;

22 }

23 @Override

24 *public* *double* volume() {

25 *return* -1;

26 }

27

28

29 }

30

31

#### **Main.java**

1 *import* java.util.Scanner;

2

3 *public* *class* Main {

4

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

6

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

8 String[] shape = *new* String[5];

9 *int* i;

10 *for*(i=0;i<5;i++){

11 String s=sc.next();

12 *if*(s.equals("Triangle"))

13 {

14 Triangle t =*new* Triangle();

15 *double* base=sc.nextDouble();

16 t.setBase(base);

17 *double* height=sc.nextDouble();

18 t.setHeight(height);

19 *double* area=t.area();

20 System.out.println("Area "+area);

21 }

22 *else* *if*(s.equals("Rectangle")){

23 Rectangle r=*new* Rectangle();

24 *double* length=sc.nextDouble();

25 *double* width=sc.nextDouble();

26 r.setLength(length);

27 r.setWidth(width);

28 *double* arear=r.area();

29 System.out.println("Area "+arear);

30 }

31 *else* *if*(s.equals("Cube")){

32 Cube c=*new* Cube();

33 *double* length=sc.nextDouble();

34 c.setLength(length);

35 *double* width=sc.nextDouble();

36 c.setWidth(width);

37 *double* height=sc.nextDouble();

38 c.setHeight(height);

39 *double* areac=c.area();

40 *double* volc=c.volume();

41 System.out.println("Area "+areac);

42 System.out.println("Volume "+volc);

43 }

44 *else* *if*(s.equals("Sphere")){

45 Sphere s1=*new* Sphere();

46 *double* radius=sc.nextDouble();

47 s1.setRadius(radius);

48 *double* area1=s1.area();

49 *double* vol=s1.volume();

50 System.out.println("Area "+area1);

51 System.out.println("Volume "+vol);

52 }

53 }

54

55 }

56

57 }

#### **Cube.java**

1 *public* *class* Cube *extends* Shape *implements* Spatial{

2

3 *private* *double* length;

4 *private* *double* width;

5 *private* *double* height;

6

7 *public* *double* getLength() {

8 *return* length;

9 }

10 *public* *void* setLength(*double* length) {

11 *this*.length = length;

12 }

13 *public* *double* getWidth() {

14 *return* width;

15 }

16 *public* *void* setWidth(*double* width) {

17 *this*.width = width;

18 }

19 *public* *double* getHeight() {

20 *return* height;

21 }

22 *public* *void* setHeight(*double* height) {

23 *this*.height = height;

24 }

25 @Override

26 *public* *double* area() {

27 *double* a=((2\*length\*width)+(2\*length\*height)+(2\*width\*height));

28 *return* a;

29 }

30 @Override

31 *public* *double* volume() {

32 *double* v=length\*width\*height;

33 *return* v;

34 }

35

36

37

38 }

39

#### **Rectangle.java**

1 *public* *class* Rectangle *extends* Shape {

2

3 *private* *double* length;

4 *private* *double* width;

5

6

7 *public* *double* getLength() {

8 *return* length;

9 }

10

11 *public* *void* setLength(*double* length) {

12 *this*.length = length;

13 }

14

15 *public* *double* getWidth() {

16 *return* width;

17 }

18

19 *public* *void* setWidth(*double* width) {

20 *this*.width = width;

21 }

22

23 @Override

24 *public* *double* area() {

25 *return* length \* width;

26 }

27

28 @Override

29 *public* *double* volume() {

30 *return* -1;

31 }

32

33 }

34

35

#### **Shape.java**

1

2 *public* *abstract* *class* Shape {

3

4 *abstract* *public* *double* area();

5 *abstract* *public* *double* volume();

6

7 }

8

#### **Spatial.java**

1

2 *public* *interface* Spatial {

3 }