11. Aim: To write Pseudo code for converting Binary to decimal

Pseudo Code:

declare the Scanner class to get the input

get the input binary number from the user.

```
While (bin > 0)
{
    a = bin % 10;
    dec+ = a * math.pow(2, i)
    bin = bin / 10
    i++
}
```

display the decimal number

12. Aim: To write Pseudo Code for Converting decimal to binary number

Pseudo Code:

declare the Scanner class for the user input

get the input decimal number from the user

declare a variable s of StringBuilder()

type

```
while (a > 0)
{
    b = a % 2
    s.append(b)
    a = a / 2;
}
```

reverse the String using reverse()

display the String

13 Aim: To write Pseudo Code to demonstrate the data types in Java

Pseudo code:

declare the variables

int a = 3;

float b = 4.56

double c = 5.6745 93276

Char d = 'H'

bool e = true

display the data types

print (a+b+c+d+e)

14. Aim: To write Java Program for

a. Implict Conversion

int intValue = 100

long longvalue = intvalue

float floatvalue = longvalue

Print Intvalue

Print longvalue

Print floatvalue

b. Explicit Conversion

```
double doublevalue = 123.45
int intvalue = int (doublevalue)

print doublevalue
print intvalue
```

c.

```
byte maxByte = 127
overflowByte = (byte) (maxByte+1)

print maxByte
print overflowByte
```

15- Aim: To write Pseudo Code for demonstrating String methods and String Constructors

PseudoCode:

```
String Constructor
  String str1 = new String("Hello world")

String methods
  String str2 = "Java";
  int length = str1.length for length of String
  Char a = str1.charAt(1); for char at index 1
  String SubStr = str2.substring(5, 11);
  String upcase = str2.toUpperCase();
  String lower = str1.toLowerCase();
```

16 Aim: To write pseudo code for printing I to
Using for loop and store in matrix 1

Pseudo Code:

Create Matrix A with dimension 1×9
Create matrix B with dimension 1×9

For i in range from 0 to 8

set matrix A[0][i] to i+1

For i in range from 9 to 1

set matrix B[0][i] to i

For i from 0 to 8

print matrix A[0][i]

for i from 9 to 1

print matrix B[0][i]

17. Aim: To write Java program for Creating a
2D array

Pseudo Code

initialize
i = 3
j = 3

For i in range from 0 to 3

For j in range from 0 to 3

matrix1[i][j] = i

using the for loop print the ... array

```
for (i = 0; i < 3; i++)
for (j = 0; j < 3; j++)
print (matrix[i][j]);
print ("\n");
```

× 18-09-2024 ×

2. Class called Car with Constructor

Pseudo Code: Input Car Output

Class Car: Procedure:-

Variables: Initialize [Variable]

String make Key:- car [String, Model [

String model Constructor [Make, Model, YEAR]

int year main (

for

Constructor //else

Input: make, model, year

Set Class variables to input variables

Method PrintCarDetails

Output: display make, model and year of the Car

Main:

Create an Object of car with make, model and year

Call printCarDetails method to display Car details

3 Bank

PseudoCode:

Class BankAccount:

int accountNumber

double balance

Constructor:

Input: accountnumber, initialBalance

Set accountNumber and balance to input Val...

Method deposit:

input: amount

If amount is less than or equal to bal...

Subtract amount from balance

else

Display "Insufficient Balance"

Main:

Create an object of BankAccount with accountnumber and initialBalance

Call deposit method to add money

Call withdraw method to take out money

4. Rectangle

PseudoCode:

Class rectangle:

Variables:

double length

double width

Constructor :

input : length, width

Set length and width to input values

Method Calculate Area :

return : length * width

Main :

Create an Object of rectangle with length and width

Call Calculate area method to get and display the area

5. Person

Pscudo Code :

Class person :

Variables :

String name

int age

Constructor :

input : name, age

Set name and age to input values

Method printPersonDetails :

Output : Display name and age

Main :

Create an object of person with name and age

Call printPersonDetails method to display the person's name and age

6. method Overriding with a Simple Calculator

Pseudocode:

```
Method add:
    input : two numbers
    return : Sum of the two numbers
method Substract:
    input : two numbers
    return : difference between the two numbers
method multiply:
    input : two numbers
    return : product of two numbers
method divide:
    input : two numbers
        if denominator is not zero:
            return : quotient
        else
            display : Can't divide by zero
Class Calculator (Inherits Calculator):
    Override method multiply :
        input : two numbers
        return : more complex calculation
main:
    Create an object of Calculator class
    Call add , Sub, multiply , and divide
```

7. Method Overloading

Class Calculator:

method Sum(int... num):

initialize total as 0

for each integers in num

Add integer to total

return total

overloaded method sum(double ... num):

Initialize total as 0.0

for each double in number:

add the double to total

return total

main:

Create an object of Calculator Class

Call sum method with a variable numbers of integers

Call overload sum method with a variable number of doubles

8. Polymorphism

Abstract Class Animal:

No implementation (to be implemented by child class)

Abstract method Sleep():

No implementation

Class Dog (Inherits Animal):

Implement Method eat():

display: "Dog is eating"

implement Method sleep():

display: "Dog is sleeping"

class Cat (Inherits Animal):

Implement method eat():

display "Cat is eating"

implement method sleep():

display: "Cat is sleeping"

Main:

Create an object of Dog class

Call eat and sleep methods

Create an object of class:

call eat and sleep methods

9. Pseudocode:

Interface Drawable:

Method draw():

no implementation

class Circle:

Implement method draw:

display: "Drawing a Circle"

Class Square:

implement method draw

display: "Drawing a Square"

main

Create an object of Circle class

Call draw method (circle)

Create an object of Square class

Call draw method (square)

10. pseudo code:

```
Class Shape:
    Variables:
        double area
        double perimeter
        method CalculateArea():
            no implementation
        method Calculateperimeter():
            No implementation
        Method getArea():
            return: area
        Method getperimeter():
            return: perimeter
    Class circle:
        Variables:
            double radius
        Constructor:
            input: radius
        Set radius and Calculate area and
                                perimeter
    implement Method CalculateArea():
        Set area to math.PI * radius * radius
```

implement method calculatePerimeter():

Set perimeter to 2 * Math.PI * radius

Class Rectangle:

Variables

double length

double width

Constructor:

input: length, width

set length, width and calculates area and perimeter

implement method calculateArea():

set area to length * width

Main:

Create an object of Circle with radius

Call calculateArea and calculatePerimeter

Create an objects of rectangle with length and width

Call calculate and calculatePerimeter method

✓

8/1/24