

## Activity - 1 (Implementing the concept)

1. Implement the concept of **Encapsulation** by creating the object of Animal.
2. How can you implement the concept of **Inheritance** in the object of Animal.
  - a. Constructor
  - b. Super keyword
  - c. This keyword

```
1 // Created a public class with private variables.
2 public class Animal
3 {
4     private String name;
5     private String type;
6
7     // Created a constructor and passed the parameters.
8     public Animal(String name, String type)
9     {
10         this.name = name;
11         this.type = type;
12     }
13
14     // Used getter-setter method to retrieve and set the data.
15     public String getName()
16     {
17         return name;
18     }
19
20     public void setName(String name)
21     {
22         this.name = name;
23     }
24
25     public String getType()
26     {
27         return type;
28     }
29
30     public void setType(String type)
31     {
32         this.type = type;
33     }
34
35
36 }
```

```

37
38 // Created a child class having a function that displays the data from the get method.
39 class Wolf extends Animal
40 {
41     private String role;
42     public Wolf(String name, String type, String role)
43     {
44         super(name, type);
45         this.role = role;
46     }
47
48     void display()
49     {
50         System.out.println(this.getName());
51         System.out.println(this.getType());
52         System.out.println(role);
53     }
54 }
55
56 // Created another child class that does the same function.
57 class Dog extends Animal
58 {
59     public Dog(String name, String type)
60     {
61         super(name, type);
62     }
63
64     void display()
65     {
66         System.out.println(this.getName());
67     }
68 }
69
70 // Created a main class where object of the classes has been created and the functions are called.
71 class Main1
72 {
73     Run | Debug
74     public static void main(String[] args)
75     {
76         Wolf wolfie = new Wolf(name: "Wolfie", type: "Mammal", role: "Wolf have been known to be guardian deity in Japanese folklore");
77         Dog dog = new Dog(name: "Dog", type: "Mammal");
78         wolfie.display();
79         dog.display();
80     }
81 }

```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\Vicer> & "C:\Program Files\Java\jre1.8.0_351\bin\java.exe" "-cp" "C:\Users\Vicer\AppData\Local\Temp\vscodesw_63336\jdt_ws\jdt.ls-java-project\bin" "Main1"
Wolfie
Mammal
Wolf have been known to be guardian deity in Japanese folklore
Dog
PS C:\Users\Vicer>

```

3. How can you implement the concept of Palindrome?

a. It should consist of knowledge that you have **pursued till now**.

b. Palindromic number/word:- That remains the same when its digits/alphabet are reversed.

i. For Example:-

1. Palindrome word:

a. eye

b. madam

c. level

d. Anna

2. Palindrome number:

a. 123

b. 1221

The sample output can be found below:

```
/usr/lib/jvm/java-11-openjdk/bin/java -ja
Please enter the word:
anna
The given word is Palendrome.

Process finished with exit code 0
|
```

Please enter the word:

hello

The given word is Not Palendrome

Process finished with exit code 0

```
1  import java.util.Scanner;
2
3  // Created a public class with a method that prints the home screen.
4  public class Palindrome
5  {
6      void display()
7      {
8          System.out.println("-----");
9          System.out.println("WELCOME TO PALINDROME CHECKER:");
10         System.out.println("-----");
11     }
12 }
13
14 // Created a child class which checks if the user input is palindromic number or not.
15 class IsPalindromeNum extends Palindrome
16 {
17     int rem;
18     int temp;
19     int sum = 0;
20     int num;
21
22     /* Stored the number temporarily and calculated the remainder then added it to the product
23     of sum and 10 and finally divided the number by 10. Finally, checked the sum and printed
24     the result accordingly.*/
25     void checkPalindromeNum()
26     {
27         temp = num;
28         while(temp>0)
29         {
30             rem = temp % 10;
31             sum = (sum*10)+rem;
32             temp /= 10;
33         }
```

```

34         if(num==sum)
35         {
36             System.out.println("The number '" + num + "' is a Palindromic Number!\n");
37         }
38
39         else
40         {
41             System.out.println("The number '" + num + "' is not a Palindromic Number!\n");
42         }
43
44     }
45 }
46
47 // Created another child class which checks if the user input is palindromic word or not.
48 class IsPalindromeChar extends Palindrome
49 {
50     String input;
51     String result;
52
53     /* Firstly, we find the midpoint of the word then store the characters from 0th index till midpoint in a variable.
54      * Then the loop is ran backwards and the reverse characters are stored in another variable.
55      * Finally, the if the length is odd or even whatever the case might be the result is printed accordingly.*/
56     void checkPalindromeChar()
57     {
58         String rev = "";
59         int mid = input.length()/2;
60         result=input.substring(0, mid);
61         for(int i = result.length() - 1; i>=0 ; i--)
62         {
63             rev += result.charAt(i);
64         }
65
66         if(input.substring((input.length() % 2 == 0 ? mid : mid + 1),input.length()).equalsIgnoreCase(rev))
67         {
68             System.out.println("The word '" + input + "' is a Palindromic Word!\n");
69         }
70         else{
71             System.out.println("The word '" + input + "' is not a Palindromic Word!\n");
72         }
73     }
74 }
75
76
77 // Created a main class in which the object of classes has been made and functions are called and the program is run.
78 class Main
79 {
80     public static void main(String[] args)
81     {
82         Scanner scanner = new Scanner(System.in);
83         IsPalindromeNum palnum = new IsPalindromeNum();
84         IsPalindromeChar palchar = new IsPalindromeChar();
85
86         // Here, depending on the user input, whether it is a number or word, the scanner identifies it and calls on the respective function.
87         palnum.display();
88         System.out.println("Enter any number/word: ");
89         if (scanner.hasNextInt())
90         {
91             palnum.num =scanner.nextInt();
92             palnum.checkPalindromeNum();
93         } else {
94             palchar.input = scanner.nextLine();
95             palchar.checkPalindromeChar();
96         }
97         scanner.close();
98     }
99 }

```

WELCOME TO PALINDROME CHECKER:  
-----  
Enter any number/word:  
nayan  
The word 'nayan' is a Palindromic Word!

PS C:\Users\Acer> & 'C:\Program Files\Java\jre1.8.0\_351\bin\java.exe' '-cp' 'C:\Users\Acer\AppData\Local\Temp\vscodesws\_63336\jdt\_ws\jdt.ls-java-project\bin' 'Main'

WELCOME TO PALINDROME CHECKER:  
-----  
Enter any number/word:  
2002  
The number '2002' is a Palindromic Number!

4. Program to find out the prime number with the concept you have learned in the class.

The sample output can be found below:

```
Please enter any number:
5
The Given Number 5 is Prime.

Process finished with exit code 0
|

Please enter any number:
8
The Given Number 8 is Not Prime.

Process finished with exit code 0
|
```

```
1 import java.util.Scanner;
2
3 // Created a public class with a method that prints the home screen
4 public class PrimeNumber {
5     void display()
6     {
7         System.out.println("-----");
8         System.out.println("WELCOME TO PRIME NUMBER CHECKER:");
9         System.out.println("-----");
10    }
11 }
12
13 // Created a child class which checks if the user input is prime or not
14 class IsPrimeNumber extends PrimeNumber
15 {
16     int num ;
17     void checkPrimeNumber()
18     {
19         int count = 0;
20
21         // If the input divided by the loop number returns zero, increment the count by 1
22         for (int i = 1; i < num; i++) {
23             if (num % i == 0)
24             {
25                 count++;
26             }
27         }
28         // Since prime number is a number that can be divided by 1 or itself, that means if the count has value greater than two it is not a prime number
29         if(count <=2)
30         {
31             System.out.println("\nThe given number '" + num + "' is a Prime Number!\n");
32         }
33         else
34         {
35             System.out.println("\nThe given number '" + num + "' is not a Prime Number!\n");
36         }
37     }
38 }
```

```

36     }
37 }
38 }
39
40 // Created a main class in which the object of classes has been made and functions are called and the program is run
41 class Main
Run | Debug
42 {   public static void main(String[] args)
43     {
44         Scanner scanner = new Scanner(System.in);
45         IsPrimeNumber pmnm = new IsPrimeNumber();
46         pmnm.display();
47         System.out.println("Enter any number: ");
48         pmnm.num = scanner.nextInt();
49         pmnm.checkPrimeNumber();
50         scanner.close();
51     }
52 }
53

```

```

-----
WELCOME TO PRIME NUMBER CHECKER:
-----

```

```

Enter any number:

```

```

7

```

```

The given number '7' is a Prime Number!

```

```

PS C:\Users\Acer>

```

## Activity - 2 (Class Diagram)

1. Draw the class diagram of all the assignments and classwork till now.

**Note: You cannot use any application to draw the diagrams. You have to draw it on a copy and click a photo or scan it.**

Note: You have to submit the PDF file with proper explanation of the code you have done and also insert the image of your code. You can take a screenshot. The file should be in report format.

Naming Convention :

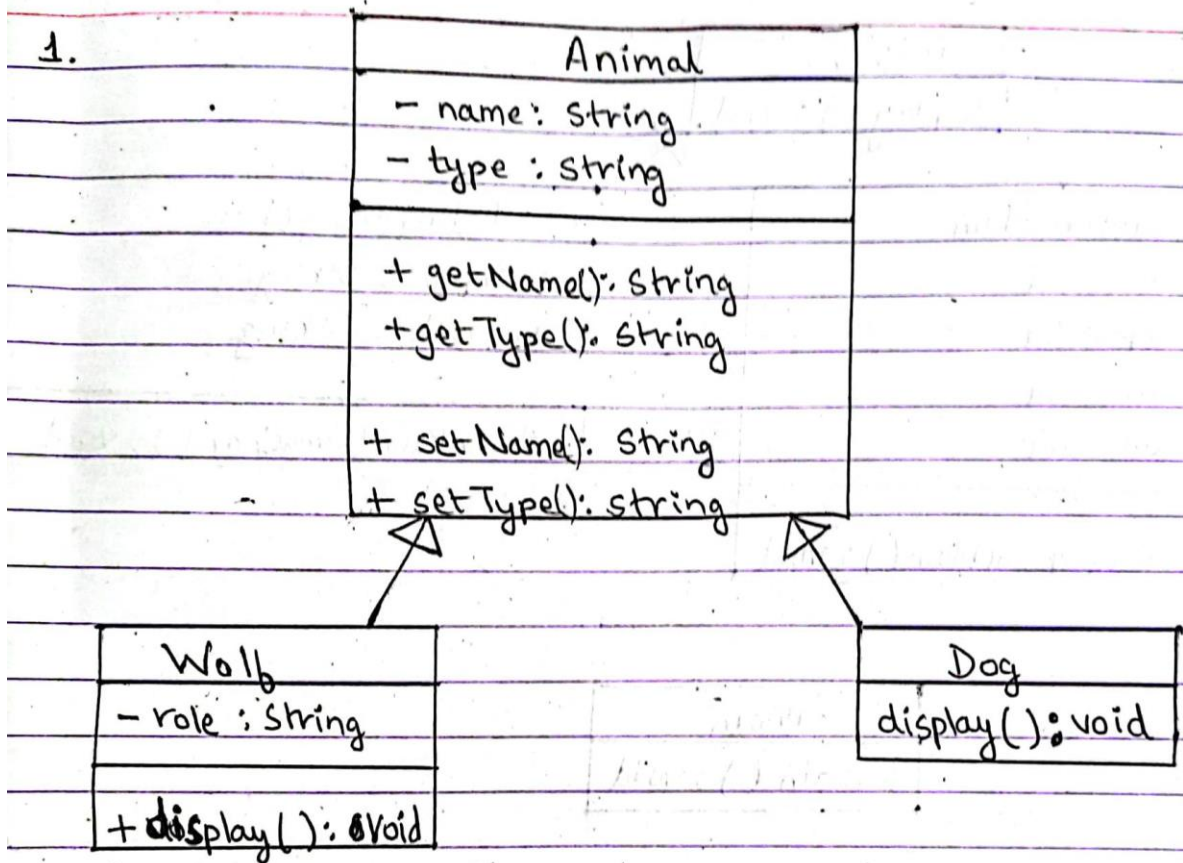
<Your-Full-Name>\_<Student-ID>\_<Group-Name>

For example:

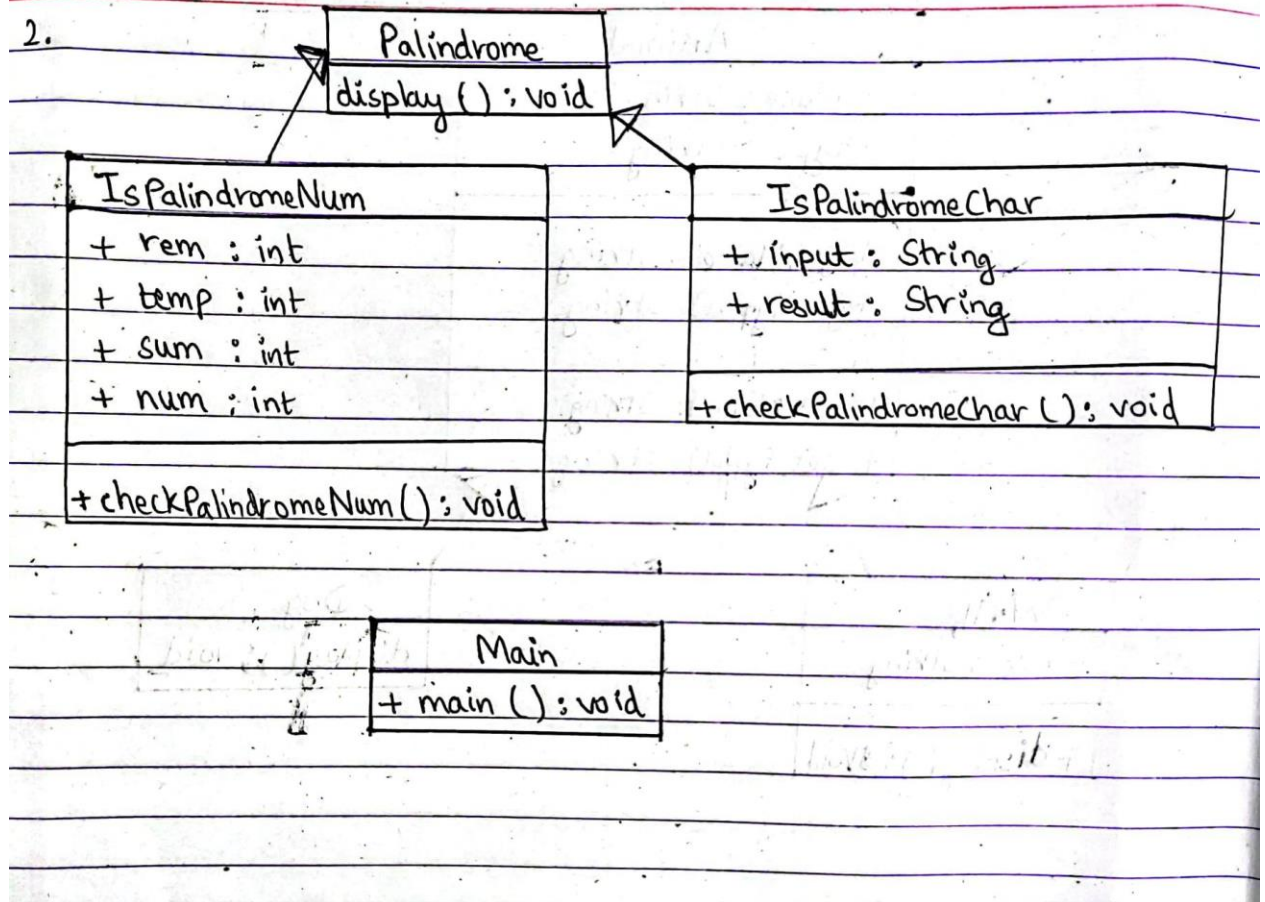
Suman-Aryal\_NP03A229068\_L5CG22



## 1. Animal:



## 2. Palindrome:



### 3. Prime Number:

