**Real Time examples of OOPs Concepts**

**1.Object**

 A real-world entity that has state and behavior is called object in java.

Example: A person has three characteristics: Identity (name), State (properties), and behavior (actions or functionality).

The state/properties of a person are black hair, black eyes, white skin, 6 feet tall, etc. The actions or behavior of the person may be “eat, sleep, walk, play, and study”.

**2.Class**

A class is basically user-defined data types that act as a template for creating objects of the identical type. It represents the common properties and actions (functions) of an object.

For example, bus and car are objects of vehicle class. Sparrow and parrot are objects of birds class. Similarly, MS Dhoni, Sachin Tendulkar, and Virat Kohli are objects of cricketer class.

**3.Encapsulation**

The process of binding data (variables) and corresponding functions together into a single unit is called [encapsulation in Java](https://www.scientecheasy.com/2020/07/encapsulation-in-java.html/).

The bag contains different stuffs like pen, pencil, notebook etc within it, in order to get any stuff you need to open that bag, similarly in java an encapsulation unit contains it's data and behavior within it and in order to access them you need an object of that unit.

**4.Abstraction**

Is a technique by which we can hide the unnecessary data that is not needed from a user and expose only that data is of interest to the user. It hides all unwanted data so that users can work only with the required data.

A realtime example of abstraction is “sending SMS”. When you need to send SMS  from your mobile, you only type the text and send the message. But you don’t know the internal processing of the message delivery.

**5.Inheritance**

It provides a mechanism for the users to reuse the existing code within the new applications.

Your car is a great example of abstraction. You can start a car by turning the key or pressing the start button. You don't need to know how the engine is getting started, what all components your car has. The car internal implementation and complex logic is completely hidden from the user.

**6.Polymorphism**

 It is a concept in which method can take more than one form based on the type of parameters, order of parameters, and number of parameters.

The best real-life example of polymorphism is human behavior. One person can have different behavior.

For example, a person acts as an employee in the office, a customer in the shopping mall, a passenger in bus/train, a student in school, and a son at home.

**Program to find magical number and neon number using inheritance**

class Siva:

def gold(warangal):

print("Deopsited 5L")

class Baby1(Siva):

def bank(warangal):

n=int(input("Enter any number:"))

if(n%2!=0):

while(n>0):

rev=0

rem=n%10

rev=rev\*10+rem

n=n/10

if(rev%2!=0):

print("Magical prime")

else:

print("Not a magical prime")

class Baby2(Siva):

def jewels(warangal):

sum=0

num=int(input("Enter any number to find neon or not:"))

square=num\*num

while(square!=0):

digit=square%10

sum=sum+digit

square=square//10

if(num==sum):

print(str(num)+"is a neon number")

else:

print(str(num)+"is not a neon number")

b=Baby1()

b.bank()

b=Baby2()

b.jewels()

**Calculating space complexity of union**

Space of a union is determined by the size of it’s largest member

#include <stdio.h>

union Example

{

int a;

double b;

}s1;

int main()

{

printf("size of union is %d",sizeof(s1));

return 0;

}

//size of union is 8

#include <stdio.h>

union Example

{

int a;

double b;

char ch[100];

}s1;

int main()

{

printf("size of union is %d",sizeof(s1));

return 0;

}

//size of union is 104

#include <stdio.h>

union Example

{

int a[5];

double b;

char ch[100];

}s1;

int main()

{

printf("size of union is %d",sizeof(s1));

return 0;

}

//size of union is 24