SCENARIO : STUDENT ENROLLMENT SYSTEM

USER STORY -- CLIENT IS RUNNING ONE EDUCATIONAL INSTITUTE NAMED ABC PVT LTD WHERE STUDENTS ENROLL IN COURSES. #CREATE A CLASS FOR THE ENROLLMENT SYSTEM WHICH TAKES THE STUDENTS DATA , HAVE THE COURSE DETAILS, ENROLL THE STUDENTS WITH PROPER COURSE , FEES AND REGISTRATION NUMBER.

ALONG WITH THESE PROCESS IT SHOULD ALSO HANDLE:

1 - COURSE NOT FOUND ERROR

2 - INVALID EDUCATIONAL BACKGROUND

3 - COURSE CAPACITY FULL

4 - OTHERS

In [5]:
```
class enrollment_system:
     def __init__(self):
        self.course={}
        self.students={}
         self.registration_count=1
```

```
    def add_course(self,course_id
        try:
            if self.course == "BSc":
                raise course(f"{self.course} is available ar
            if self.cours == "B.com":
                raise course(f"{self.course} is available ar
            if self.course=="BCA":
                raise course(f"{self.course} is available ar
            self.choose_course=self.choose_course.get(self.c
        except course as c:
            print(c)
        except course as c:
            print(c)
        except course as c:
            print (c)
        finally:
            print("congralations")
```

In [8]:
```
a =enrollment_system()
```

In [11]:
```

```

Out[11]:
```
'BCA'
```

In [ ]:
```

```

In [ ]:
```
course = input("enter the course name:")
if course == "Datascience":
    print(f"{course} is available.\nThe duration of course i
if course == "Java":
    print(f"{course} is available.\nThe duration of course i
if course == "Python":
    print(f"{course} is available.\n The duration is 45 days
else:
    print(f"{course} is not listed")
```

In [ ]:
```
name = []
domain = []
def extract_mail():
    email = input("Enter the mail id")
    split = email.split("@")
    name.append(split[0])
```

```
    domain.append(split[1])
    return name , domain
```

In [ ]: