```
from enum import Enum

class CoverType(Enum):
    """ Enum Class for Cover type"""
    FULL = 1
    HALF = 2
    NONE = 3

class InsuranceCover:
    """Class to represent insurance cover"""

    # Initializer
    def __init__(self, coverType:CoverType, companyName="", companyEmail=""):
        self.__coverType = coverType
        self.__companyName = companyName
        self.__companyEmail = companyEmail

    # Setter and Getter functions
    def setCoverType(self, cover:CoverType):
        self.__coverType = cover

    def getCoverType(self):
        return self.__coverType

    def setCompanyName(self, name):
        self.__companyName = name

    def getCompanyName(self):
        return self.__companyName

    def setCompanyEmail(self, email):
        self.__companyEmail = email

    def getCompanyEmail(self):
        return self.__companyEmail

    # Del function
    def __del__(self):
        print("Deleting an instance of the class Insurance Cover")

```

```
from enum import Enum

class ChargeType(Enum):
    """Enum type class for chargeType"""
    Consultation = 1
    Nursing = 2
    Hospital = 3
    Pharmacy = 4
    RoomCharges = 5

class Patient:
    """Class to represent a Patient"""

    # Initializer
    def __init__(self, insuranceCover, patientName="", phoneNumber="", emailAddress=""):
        self.__patientName = patientName
        self.__phoneNumber = phoneNumber
        self.__emailAddress = emailAddress
        self.__insuranceCover = insuranceCover

    # Setter and Getter functions
    def setName(self, name):
        self.__patientName = name

    def getName(self):
        return self.__patientName

    def setPhoneNumber(self, number):
        self.__phoneNumber = number

    def getPhoneNumber(self):
        return self.__phoneNumber

    def setEmail(self, email):
```

```
        self.__emailAddress = email

    def getEmail(self):
        return self.__emailAddress

    # Function to get Cover type from the insurance cover
    def getInsuranceCover(self):
        return self.__insuranceCover.getCoverType()

    # Del function
    def __del__(self):
        print("Deleting an instance of the class Patient")

class Charge:
    """Class to represent the charge type and amount"""

    # Initializer
    def __init__(self, chargeType: ChargeType, amount: float):
        self.__chargeType = chargeType
        self.amount = amount

    # Setter and getter functions
    def setChargeType(self, charge: ChargeType):
        self.__chargeType = charge

    def getChargeType(self):
        return self.__chargeType

    def __del__(self):
        print("Deleting an instance of the class Charge")
```

```
class Hospital:
    """Class to represent hospital"""

    # Initializer
    def __init__(self, hospitalId="", hospitalName="", hospitalEmail=""):
        self.__hospitalId = hospitalId
        self.__hospitalName = hospitalName
        self.__hospitalEmail = hospitalEmail
        self.__totalAmount = 0

    # Setter and Getter functions
    def setHospitalName(self, name):
        self.__hospitalName = name

    def getHospitalName(self):
        return self.__hospitalName

    def setHospitalID(self, hospitalId):
        self.__hospitalId = hospitalId

    def getHospitalID(self):
        return self.__hospitalId

    def setHospitalEmail(self, email):
        self.__hospitalEmail = email

    def getHospitalEmail(self):
        return self.__hospitalEmail

    # Function to generate Hospital Bill
    def generateBill(self, patient, chargeList, taxPercent):
        return MedicalBill(chargeList, taxPercent, patient)

    # Del function
    def __del__(self):
        print("Deleting an instance of the class Hospital")
```

```
class Bill:
    """Class representing a Bill"""

    # Initializer
    def __init__(self, billNumber: float, recipientId=""):
        self.__billNumber = billNumber
```

```
        self.__recipientId = recipientId

    # Setter and Getter Functions
    def setBillNumber(self, billNumber):
        self.__billNumber = billNumber

    def getBillNumber(self):
        return self.__billNumber

    def setRecipientId(self, recipientID):
        self.__recipientId = recipientID

    def getRecipientId(self):
        return self.__recipientId

    def printBill(self):
        print("Bill Details")

    # Del function
    def __del__(self):
        print("Deleting an instance of the class Bill")
```

```
class MedicalBill(Bill):
    """Class to represent MedicalBill"""

    # Initializer
    def __init__(self, charges: list[Charge], taxPercentage: float, patient: Patient):
        super().__init__(123, patient.getEmail())
        self.__charges = charges
        self.__taxPercentage = taxPercentage

        # Calculating the total Amount
        self.__totalAmount = 0
        for charge in self.__charges:
            self.__totalAmount += charge.amount

        if patient.getInsuranceCover() == CoverType.FULL:
            self.__totalAmount = 0
        elif patient.getInsuranceCover() == CoverType.HALF:
            self.__totalAmount = self.__totalAmount / 2

        self.__totalAmount += (self.__totalAmount * self.__taxPercentage) / 100

    # Setter and getter functions
    def setCharges(self, charges):
        self.__charges = charges

    def getCharges(self):
        return self.__charges

    def getTotalAmount(self):
        return self.__totalAmount

    # Print Bill method
    def printBill(self):
        """
        This method will contain the logic to calculate, format and print the entire medical bill
        """
        pass

    # Del function
    def __del__(self):
        print("Deleting an instance of the class MedicalBill")
```