**Q1**. Can two different unrelated classes in a single program contain functions with the same name and parameter signature?

**Answer: Yes. Since the naming scope of two unrelated classes are different in a program.**

**Q No. 2**

ABC University is a large institution with several campuses. Each campus has a different name, address, and distance to the city center.

The University consists of a number of faculties, such as the Art Faculty, the Science Faculty, and so on. Each faculty has a name, dean and building. ABC University offers different programmes and each programme has a code (valid format: P-nnn, where n can be any digit), title, level and duration (1, 2 or 3). Each programme comprises several courses, different programmes have different courses. Each course has a code (valid format C-nnn, where n can be any digit) and course title.

Each of the students is enrolled in a single programme of study which involves a fixed set of core courses specific to that programme as well as a number of electives taken from other programmes. Students work on courses and are awarded a grade in any course if he/she passes the course. Otherwise the student has to re-take the failed course. The system needs to record the year and term in which the course was taken and the grade awarded to the student. Every student has a unique ID. The system also keeps the student name, birthday and the year he/she enrolled in the course.

1. Identify all the classes and objects that will be needed.

**Campus**

**Faculty**

**Programme**

**Course**

**Result**

**Student**

1. Identify the key attributes and functionalities of each class.

**Campus**: name, address, distance

**Faculty**: name, dean, building

**Programme**: code, title, level, duration, const courses[a,b,c…], **validateCode(string){\*/WORKING\*/}, validateDuration(int){ \*/WORKING\*/}**

**Course:** code, title, **validateCode(string){ \*/WORKING\*/}**

**Result**: year, term, grade

**Student**: ID, name, birthday, enrollmentYear

1. Explain how Object Oriented approach can make your program more robust, secure and efficient with reference to the given scenario.

**[FREE RESPONE] each of the above entity’s data (instance) is contained within the objects, making it hidden from accidental changes and access from outside the object. For example student1 cannot access the data contained in student2 instance. OR ANY OTHER OOP approach**

**Q No. 3 [5 x 2 = 10 Points]**

**a.** Consider the code snippet given below, identify and explain the error and also provide the corrected code.

class Employee

{

string name;

set\_Name(string a)

{

name = a;

}

};

int main()

{

Employee e1;

e1.set\_Name("Rehan");

}

**#include<iostream>**

**using namespace std;**

**class Employee**

**{**

**string name;**

**set\_Name(string a)**

**{**

**name = a;**

**}**

**public:**

**my\_func(string a)**

**{**

**set\_Name(a);**

**}**

**};**

**int main()**

**{**

**Employee e1;**

**//e1.set\_Name("Rehan");**

**e1.my\_func("Rehan");**

**}**

**b. b. identify the error in following code**

**#include<iostream>**

**using namespace std;**

**class Student{**

**int \*rollnumber;**

**public:**

**Student(int arollnumber){**

**rollnumber = new int;**

**\*rollnumber = arollnumber;**

**}**

**int display(){**

**cout<<\*rollnumber<<endl;**

**}**

**void deleteRollNumber(){**

**delete rollnumber;**

**}**

**~Student(){**

**delete rollnumber;**

**}**

**};**

**int main(){**

**Student s(12);**

**s.deleteRollNumber();**

**}**

**The solution discussed in class.**