

Assignment 4 Sample Test Case

Question 1

```
public static void testCourse() {  
    // Since we cannot create any Course object, the testCourse method of the Course  
    class must be empty.  
}
```

Question 2

```
public static void testMajorRequired() {  
    // Since we cannot create any MajorRequired object, the testMajorRequired method of  
    the Course class must be empty.  
}
```

Question 3

```
public static void testMajorElective() {  
    // Since we cannot create any MajorRequired object,  
    // the testMajorElective method of the Course class must be empty.  
}
```

Question 4

```
public static void testBase() {  
    Base b = new Base("DS1001", "EntryCourse");  
    System.out.println(b.getCode() == "DS1001");  
    System.out.println(b.getTitle() == "EntryCourse");  
    System.out.println(b.getPreRequisite() == b);  
}
```

Question 5

```
public static void testCourse() {  
    // Since we cannot create any Course object, the testCourse method of the Course  
    class must be empty.  
}
```

```
public static void testBase() {  
    Base b = new Base("DS1001", "EntryCourse");  
    System.out.println(b.getCode() == "DS1001");  
    System.out.println(b.getTitle() == "EntryCourse");  
    System.out.println(b.getPreRequisite() == b);  
}
```

```
public static void testMajorRequired() {  
    // Create a Base object  
    Base b = new Base("DS1001", "EntryCourse");  
    // Create a first MajorRequired object with the base course as pre-requisite  
    MajorRequired mr1 = new MajorRequired("DS200X", "OOP", b);  
}
```

```

    // Create a second MajorRequired object with the first major required course as
    pre-requisite
    MajorRequired mr2 = new MajorRequired("DS300X", "Data Mining", mr1);
    // Test mr1
    System.out.println(mr1.getCode() == "DS200X");
    System.out.println(mr1.getTitle() == "OOP");
    System.out.println(mr1.getPreRequisite() == b);
    // Test mr2
    System.out.println(mr2.getCode() == "DS300X");
    System.out.println(mr2.getTitle() == "Data Mining");
    System.out.println(mr2.getPreRequisite() == mr1);
}

public static void testMajorElective() {
    // Create a Base object
    Base b = new Base("DS1001", "EntryCourse");
    // Create a first MajorRequired object with the base course as pre-requisite
    MajorRequired mr1 = new MajorRequired("DS200X", "OOP", b);
    // Create a MajorElective object with the major required course as pre-requisite
    MajorElective me1 = new MajorElective("DS300X", "Data Mining", mr1);
    // Test mr1
    System.out.println(mr1.getCode() == "DS200X");
    System.out.println(mr1.getTitle() == "OOP");
    System.out.println(mr1.getPreRequisite() == b);
    // Test mr2
    System.out.println(me1.getCode() == "DS300X");
    System.out.println(me1.getTitle() == "Data Mining");
    System.out.println(me1.getPreRequisite() == mr1);
}

```

Question 6

```

public static void testManyCourses() {
    Base b = new Base("DS1001", "EntryCourse");
    MajorRequired mr1 = new MajorRequired("DS200X", "OOP", b);
    MajorElective me1 = new MajorElective("DS300X", "Data Mining", mr1);
    ManyCourses mc = new ManyCourses();
    mc.addCourse(b);
    mc.addCourse(mr1);
    mc.addCourse(me1);
    mc.listCourses();
}

```