|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task No. | 2 | Task Title | | Practical Tasks | | | | Due Date | 5/04/2019 | |
| Unit / Module details | | | | | | | | | | |
| Qualification/Course | | | Code | ICT60515 | Title | Advanced Diploma of Computer Systems Technology | | | | |
| Unit/Module | | | Code | ICTPRG527 | Title | Apply Intermediate Object-Oriented Language Skills | | | | |
| Assessment details | | | | | | | | | | |
| Assessor | | | Assessor’s name | | Vikram Behal | | Assessor’s phone | | | 03 9354 6130 |
|  | | | Assessor’s email | | vbehal@kangan.edu.au | | | | | |
| Elements / Learning outcomes | | | 1,2,3,4,5,6 | | | | | | | |
| Task overview*(see below for specific instructions and criteria)* | | | This assessment item requires students to work in individually and accomplish their task professionally and in time. It is their responsibility to inform/contact teacher if they need any clarification to accomplish this task. This assessment item is open book and they can use any available resources. This assessment item is practical task and students need to complete all practical tasks as per Assessment item Description (At the end of this document). | | | | | | | |
| Assessment conditions | | | * This information is to be handed to each student to outline the assessment requirements. * This is open book assessment answer all questions. * This Assessment Item requires students to work individually and address all requirements as per Assessment item Description * Students are required to accomplish their task professionally and in time * Students must submit this Assessment item at MyLearning * This assessment item is classroom based * This assessment item is due in week 7 * Student can negotiate time with teacher under special circumstances * Read the Plagiarism policy and procedure carefully to understand the consequences that you could face if your work is plagiarized. * You plagiarize if you present other writers' words or ideas as your own. * You must use citations to document all ideas and significant information that are not your own. * Submit this document by the due date (penalties applies) * Insert your name and student id in the footer. * Use Font: Arial Font Size: 10 Heading: Bold Arial Heading Size: 12 Sub Heading: Bold Arial Sub Heading Size: * Marking Criteria for this assessment Item is as under | | | | | | | |
| Resources | | | Resources are available through MyLearning | | | | | | | |
| How to submit | | | Complete Assessment items as per assessment Item description and upload assessment item at MyLearning  All items submitted must be clearly marked with the following details:  • Your full name  • Your student number  • Your class group  • The date  This cover sheet must accompany all items submitted. | | | | | | | |

| Assessment instructions and criteria | | | |
| --- | --- | --- | --- |
| With competency based assessment **all** assessment requirements for each assessment task must be assessed as satisfactory **(S)** for a competent **(CA)** result to be recorded. If an assessment result for any task is assessed as not satisfactory **(NS)** a resubmission will be required for the outstanding **(NS)** assessment task.  Rows can be deleted by left clicking to the left of the row and click *Backspace*  Rows can be added by left clicking to the left of the row and click *CTRL+C* and then *CTRL+V* | | | |
| **Instruction/Task** | **Criteria that you will be assessed on** | **S** | **NS** |
| Student need to complete all practical tasks as per assessment Item Description | Student has completed all practical tasks as per assessment Item Description | ☐ | ☐ |
| Student need to check all requirements as per assessment item check list | Student has checked all requirements as per assessment item check list | ☐ | ☐ |
| Student need to review all programs requirements | Student has reviewed all programs requirements | ☐ | ☐ |
| Student need to divide multiple source-code files into logical units and package | Student has divide multiple source-code files into logical units and package | ☐ | ☐ |
| Student need to use programming language for storage of collections of data | Student has used programming language for storage of collections of data | ☐ | ☐ |
| Student need to use programming language for sorting and searching facilities | Student has used programming language for sorting and searching facilities | ☐ | ☐ |
| Student need to use eclipse to make files to automate program building | Student has used eclipse to make files to automate program building | ☐ | ☐ |
| Student need to use the operator and function/method overloading facilities available in the language | Student has used the operator and function/method overloading facilities available in the language | ☐ | ☐ |
| Student need to demonstrate ability to use exception-handling techniques to ensure program stability | Student has demonstrated ability to use exception-handling techniques to ensure program stability | ☐ | ☐ |
| Student need to demonstrate use of a class | Student has demonstrated use of a class | ☐ | ☐ |
| Student need to design and implement programs that use the language facilities to insert, update and delete data stored in a database | Student has designed and implemented programs that use the language facilities insert, update and delete data stored in a database | ☐ | ☐ |
| Student need to implement primary keys and built relationships at database level to maintain transactional integrity | Student has implemented primary keys and built relationships at database level to maintain transactional integrity | ☐ | ☐ |
| Student need to demonstrate use of eclipse, WAMP | Student has demonstrated use of eclipse, WAMP | ☐ | ☐ |
| Student need to use debugger to detect logical and coding errors | Student has used debugger to detect logical and coding errors | ☐ | ☐ |
| Student need to test their code to ensure it complies with program specification | Student has tested their code to ensure it complies with program specification | ☐ | ☐ |
| Student need to captur and documented test results | Student has captured and documented test results | ☐ | ☐ |
| Student need to create design document to create program | Student has created design document to create program | ☐ | ☐ |
| Student need to read, understand and interpret created design document to create code | Student has read, understood and interpreted created design document to create code | ☐ | ☐ |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Student details | | | | | | | | | | | |
|  | Student’s name | | | *Benjamen Calleja* | | Student’s ID | | | Cal14385330 | | |
| Unit/Module details | | | | | | | | | | | |
| Unit/Module | Code | | ICTPRG527 | Title | Apply Intermediate Object-Oriented Language Skills | | | | | | |
|  | Task title | | Practical Tasks | | | | | | | | |
| Assessment feedback / comments | | | | | | | | | | | |
| Click here to enter text. | | | | | | | | | | | |
| Re-submission (where applicable) | | | | | | | | | | | |
| Unsatisfactory tasks are to be re-submitted according to the details below. Type NA if not applicable. | | | | | | | | | | | |
| **Task (or component of task) to be re-submitted** | | **Additional evidence required** | | | | | **Re-Submission date** | | | **S** | **NS** |
| Task for re-submission | | Evidence | | | | | Date | | |  |  |
| Task for re-submission | | Evidence | | | | | Date | | |  |  |
| Task for re-submission | | Evidence | | | | | Date | | |  |  |
| Result for this assessment task | | | | | | | | | | | |
|  | Your final result for this unit will be provided on the ***Unit Result Record*** at completion of all assessment tasks | | | | | | | | | | |
|  | Result | | | Choose an item. | | | | | | | |
|  | Assessor’s signature | | |  | | | **Date** |  | | | |

**Practical Task 1 (Recommended completing in week 4)**

1. Students are required to create a java Program which will create an array as under and sort it in ascending order. Make sure array must have following five elements.

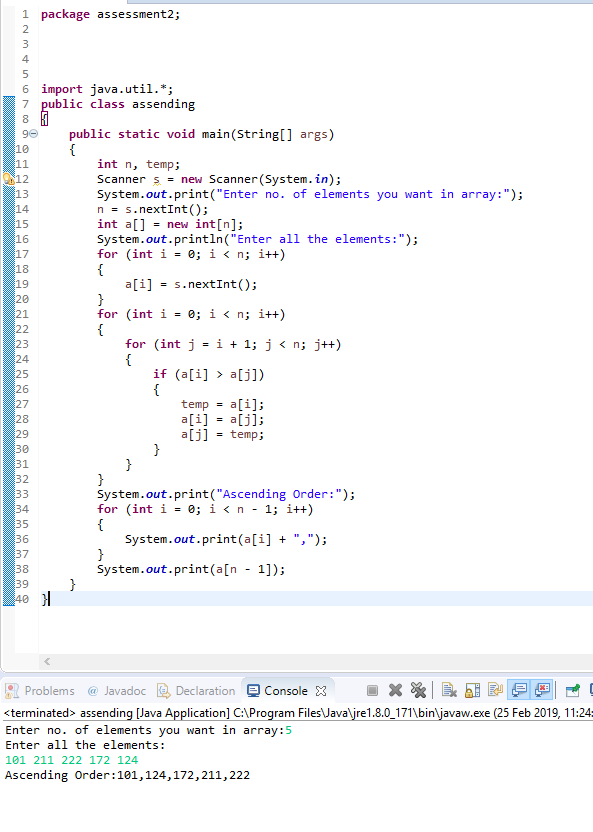
a[0] = 101;

a[1] = 211;

a[2] = 222;

a[3] = 172;

a[4] = 124;



1. Students are required to create a java Program which will create an array as under and allow user to input an integer. Once user will enter integer, program must search integer in the array and display message accordingly. If user find the number in array they must display “they found number” else it will display “ Number not found”. Make sure array must have following five elements.

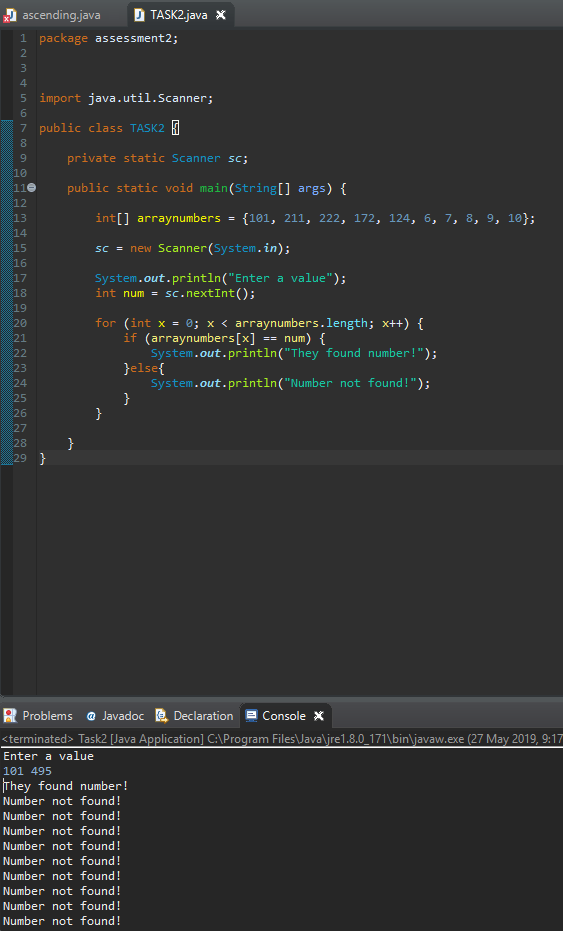
a[0] = 101;

a[1] = 211;

a[2] = 222;

a[3] = 172;

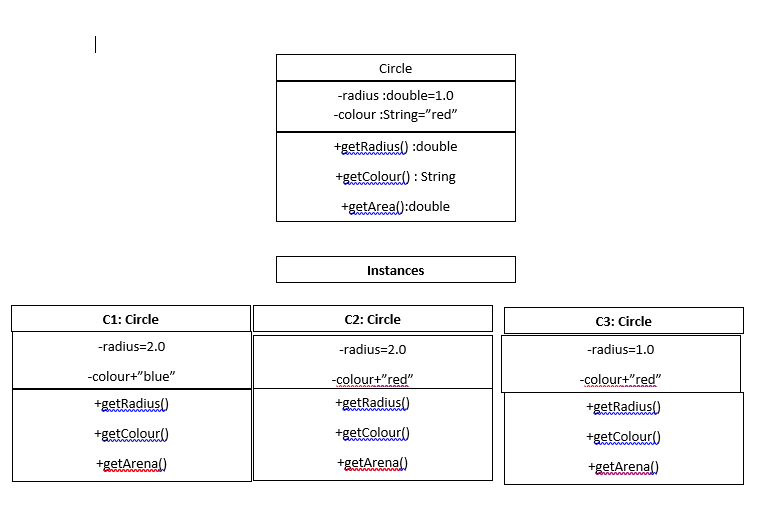
a[4] = 124;



**Practical Task 2 (Recommended completing in week 5)**

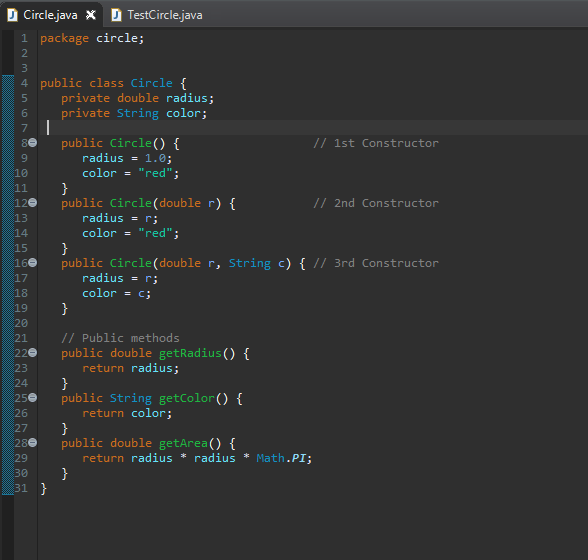
1. Students are required to create UML Diagrams for class MyCircle (For Circle we need radius as instance variable). Add appropriate methods for class MyCircle.

[ Create UML Diagram here ]



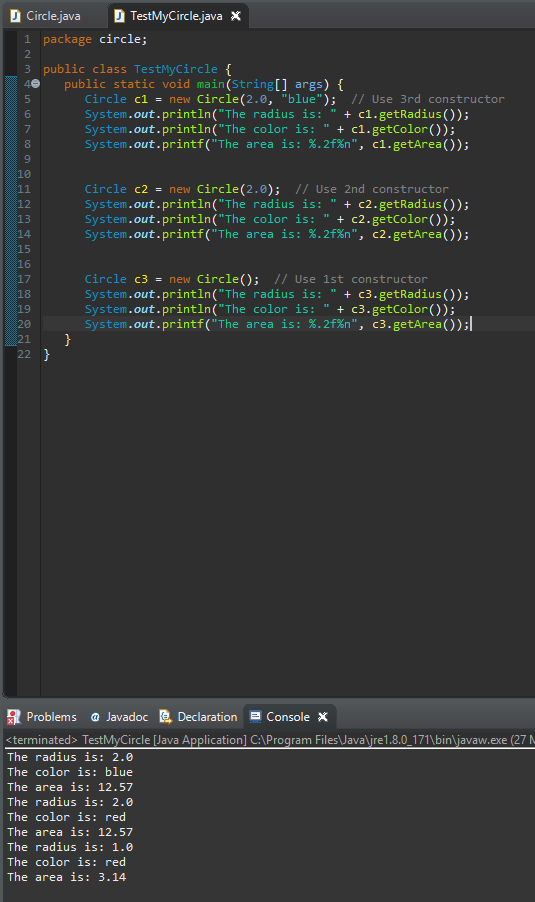
1. Students are required to create java class for above UML Diagrams(For Circle we need radius as instance variable). Students must use eclipse to create this class.
   1. Students are required to create appropriate instance variable.
   2. Students are required to create appropriate constructors.
   3. Students are required to create all set methods.
   4. Students are required to create all get methods
   5. Students are required to create one method to return area of circle. Formula to calculate area is as under:

Area = PI \* radius \* radius

* 1. Students can create any additional method.  
     

1. Students are required to create java class “TestMycircle” to test above class (MyCircle). Students must use eclipse to create this class. Formula to calculate area is as under:

Area = PI \* radius \* radius



1. If student found any issue in their class they must mention issue found and how they have resolved these issues.
2. Create a user documentation for class MyCircle and TestMyCircle

**UML notations. A class is represented as a 3-compartment box, containing name, variables, and methods, respectively. Class name is shown in bold and centralized. An instance is also represented as a 3-compartment box, with instance name shown as instanceName:Classnameand underlined.**

1. **A *class* is a programmer-defined, abstract, self-contained, reusable software entity that mimics a real-world thing.**
2. **A class is a 3-compartment box containing the name, variables and the methods.**
3. **A class encapsulates the data structures (in variables) and algorithms (in methods). The values of the variables constitute its *state*. The methods constitute its *behaviors*.**
4. **An *instance* is an instantiation (or realization) of a particular item of a class.**

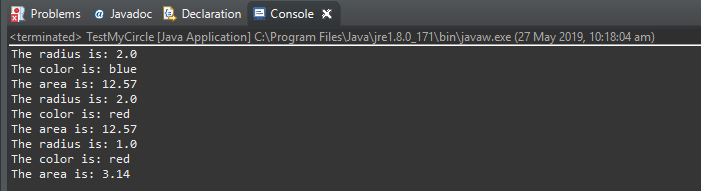
**A class can be visualized as a three-compartment box, as illustrated:**

1. ***Name* (or identity): identifies the class.**
2. ***Variables* (or attribute, state, field): contains the *static attributes* of the class.**
3. ***Methods* (or behaviors, function, operation): contains the *dynamic behaviors* of the class.**

**In other words, a class encapsulates the static attributes (data) and dynamic behaviors (operations that operate on the data) in a box.**

1. Add screen print of TestmyCircle here.

[Add screen print of output of TestMycircle here]



**Practical Task 3 (Recommended to complete in week 6)**

1. Students are required to SQL File/script to do following:
   1. Create one database called students
   2. Create user 'user2'@'localhost' password 'Password01' and GRANT him permissions like SELECT, INSERT, UPDATE and DELETE.
   3. CREATE TABLE student as under (sid, name and lastname are three fields in this table):

sid VARCHAR(10) PRIMARY KEY,

name VARCHAR(30),

lastname VARCHAR(30)

* 1. CREATE TABLE unit as under (uid and uname are fields):

uid VARCHAR(10) PRIMARY KEY,

uname VARCHAR(30)

* 1. CREATE TABLE enrolled as under with following fields:

studentid VARCHAR(10),

unitid VARCHAR(10),

year INT,

sem INT,

fees int,

PRIMARY KEY (studentid, unitid, year, sem),

FOREIGN KEY (studentid) REFERENCES student(sid)

* 1. ALTER TABLE enrolled ADD FOREIGN KEY unitid and link it with uid of unit (students database) as under:

(unitid) REFERENCES unit(uid);

* 1. Inset at least six records in all tables
  2. Import script in your WAMP make sure you logged in as root in WAMP.
  3. Upload SQL file/script with this file

[ add screen print of output of your import here and upload SQL file created by you with this document]

use students;

create user 'user2'@'localhost' identified by 'Password01';

grant SELECT, INSERT, UPDATE, DELETE on students TO 'user2'@'localhost' identified by 'Password01';

CREATE TABLE student(

sid VARCHAR(10) PRIMARY KEY,

fname VARCHAR(30),

lname VARCHAR(30));

CREATE TABLE unit(

uid VARCHAR(10) PRIMARY KEY,

uname VARCHAR(30));

CREATE TABLE enrolled(

studentid VARCHAR(10),

unitid VARCHAR(10),

year INT,

sem INT,

fees int,

PRIMARY KEY (studentid, unitid, year, sem),

FOREIGN KEY (studentid) REFERENCES student(sid));

ALTER TABLE enrolled ADD FOREIGN KEY (unitid) REFERENCES unit(uid);

INSERT INTO student VALUES('S001', 'Delia','Johnson');

INSERT INTO student VALUES('S002', 'Ava','Young');

INSERT INTO student VALUES('S003', 'Mary','Thomas');

INSERT INTO student VALUES('S004', 'Shawn','Ward');

INSERT INTO student VALUES('S005', 'Patricia','Stephenson');

INSERT INTO student VALUES('S006', 'Willie','Carr');

INSERT INTO unit VALUES('U001','Computer Science');

INSERT INTO unit VALUES('U002','Maths');

INSERT INTO unit VALUES('U003','Science');

INSERT INTO unit VALUES('U004','English');

INSERT INTO unit VALUES('U005','Information Technology');

INSERT INTO unit VALUES('U006','Psychology');

INSERT INTO enrolled VALUES ('S001','U004',2010,1,1000);

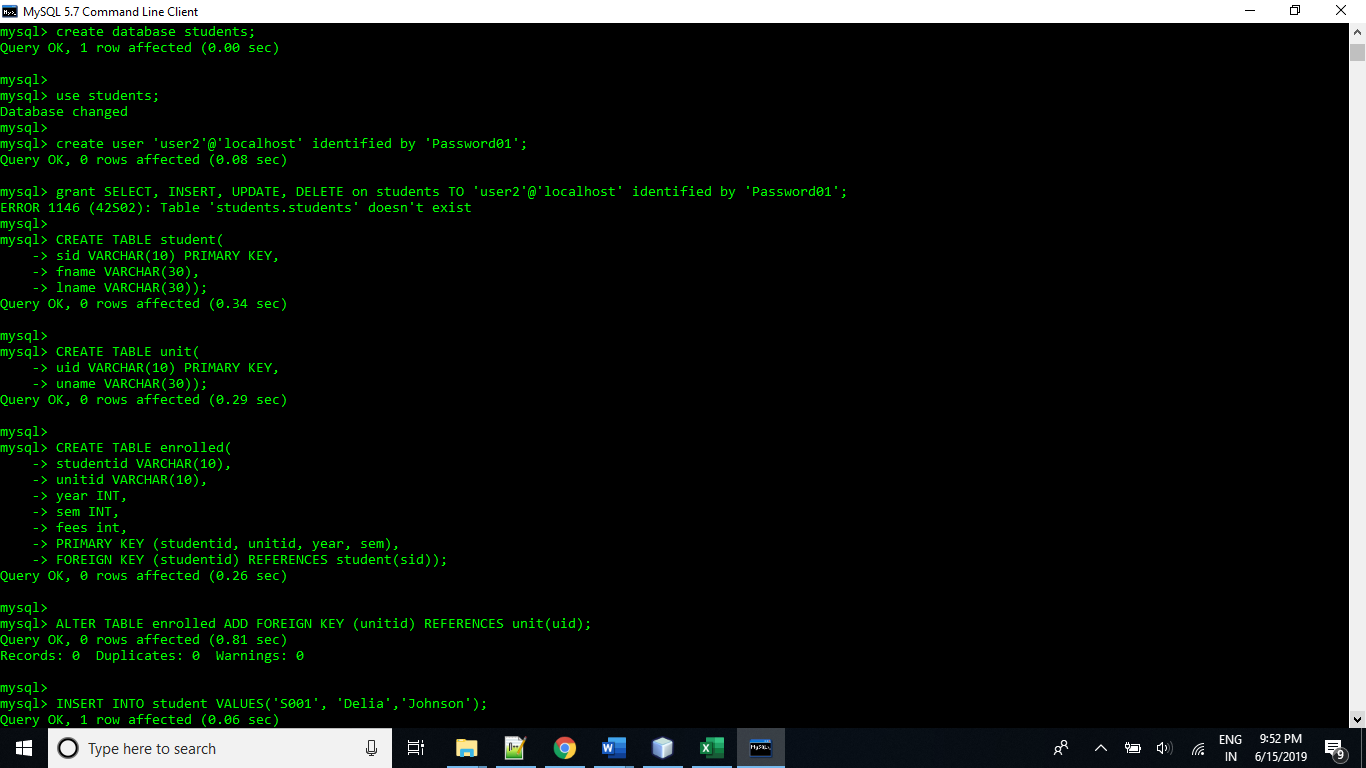
INSERT INTO enrolled VALUES ('S004','U003',2015,2,12000);

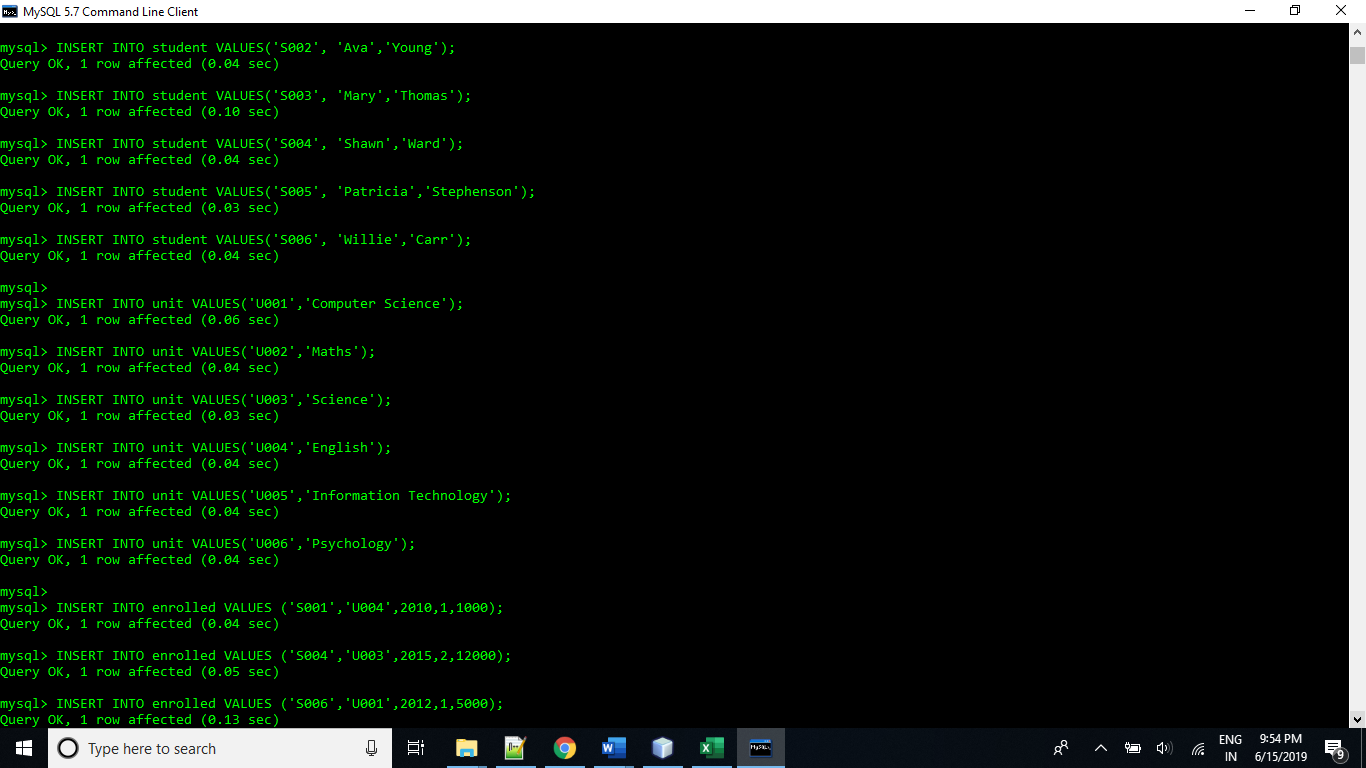
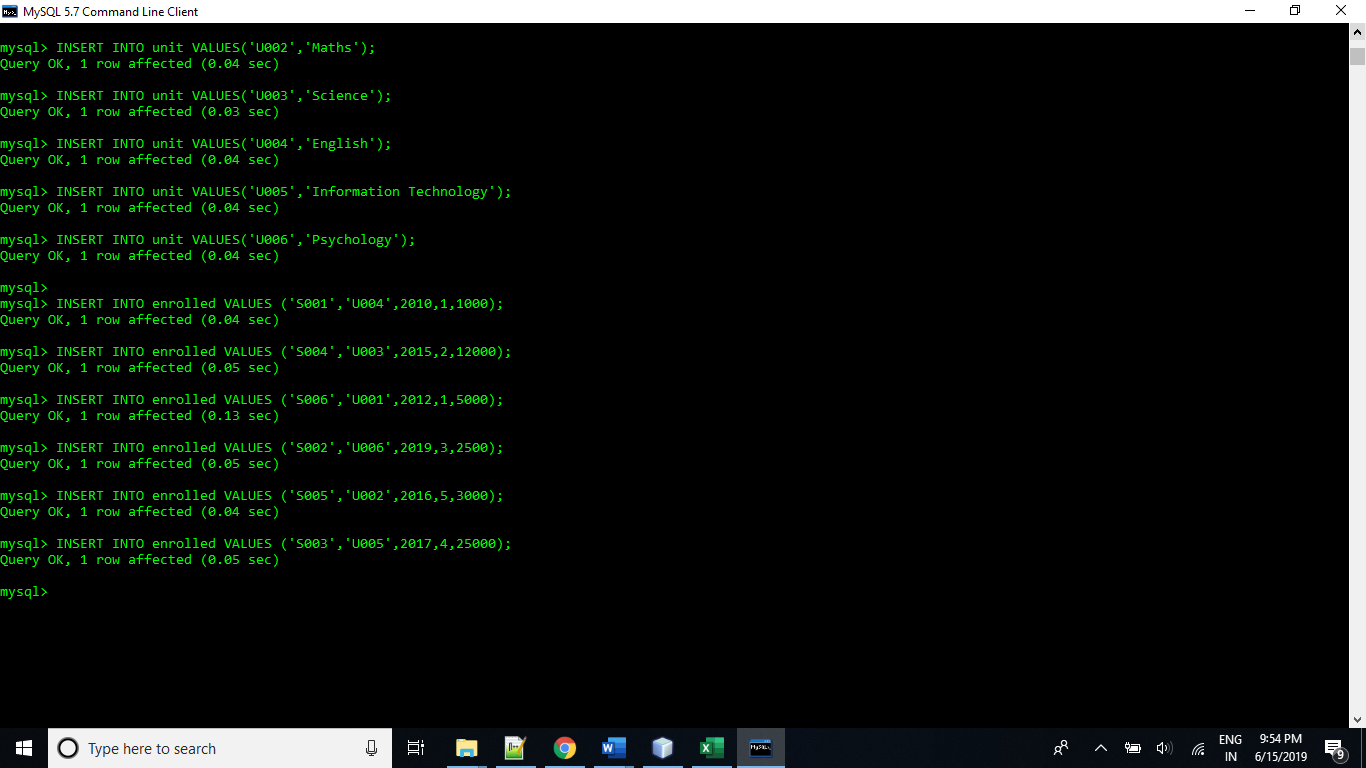
INSERT INTO enrolled VALUES ('S006','U001',2012,1,5000);

INSERT INTO enrolled VALUES ('S002','U006',2019,3,2500);

INSERT INTO enrolled VALUES ('S005','U002',2016,5,3000);

INSERT INTO enrolled VALUES ('S003','U005',2017,4,25000);



1. Students are required to create a sql file to insert Following (two) record in student table:
   1. ’V1234', 'Vikram','Teacher’ and ‘M1234', 'Murad’,'Quazi’),
   2. Insert 4 more records in student table

[ add screen print of student table here]

INSERT INTO student VALUES('V1234','Vikram','Teacher');

INSERT INTO student VALUES('M1234', 'Murad','Quazi');

INSERT INTO student VALUES('A1234','Maria','Higginbotham');

INSERT INTO student VALUES('B1234', 'Mildred','Hamilton');

INSERT INTO student VALUES('C1234','Sarah','Vera');

INSERT INTO student VALUES('D1234', 'Louis','Woolard');

