



UNIVERSITI TEKNOLOGI MARA
KEDAH BRANCH
SCHOOL OF INFORMATION SCIENCE
COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

DIPLOMA IN LIBRARY INFORMATIC (IM144)

IML 208: Programming For Libraries

ASSESSMENT 1: Programming

MUET TEST FOR REPEATER

Prepared by:

MAIZATUL NURAIN SYAZWANA BINTI MOHD AKHIR (2022807678)

GROUP KIM1443E.

Prepared for:

SIR AIRUL SHAZWAN BIN MOHD NORSHAHIMI

Submission date:

4th JANUARY 2024

MUET TEST FOR REPEATER

PREPARED BY:

MAIZATUL NURAIN SYAZWANA BINTI MOHD AKHIR (2022807678)

GROUP KIM1443E

IML 208 – DIPLOMA IN INFORMATIC LIBRARY

SCHOOL OF INFORMATION SCIENCE

COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS

UNIVERSITI TEKNOLOGI MARA (UITM)

KEDAH BRANCH

TABLE OF CONTENT

INTRODUCTION	1
SNAPSHOT OF CODE.....	2-4
SNAPSHOT OF GUI.....	4
SNAPSHOT OF DATABASE.....	5
FLOWCHART.....	6

Introduction:

Do you know what MUET stands for? This acronym, which pre-university students are particularly familiar with, stands for The Malaysian University English Test. However, the Malaysian government developed this English exam to provide a standard for English proficiency among students in the country. Thus, continue reading if you intend to pursue your degree at a nearby university because it contains all the information you require.

MUET was previously held only twice a year, in the middle of the year between April and May, and at the end of the year between October and November. However, as of 2012, MUET is now held three times a year in March, July, and November.

If you did not pass the first time around or are not happy with your score, you have the option to retake the MUET. There is no upper limit to this, but students would be well advised to make advance plans. Students need to be aware of the local university's admissions season. This is due to the fact that enrolment in degree programs at nearby universities is restricted to MUET-trained students alone. Students who have completed their matriculation or STPM coursework are still eligible to take the MUET. However, in order for students to register for their selected courses as soon as the registration for local university applications opens, it is best to receive both STPM/matriculation results along with MUET results.

Snapshot of CODE:

```
File Edit Selection View Go Run Terminal Help
registration.py X
registration.py > collect_data
1 import tkinter as tk
2 import mysql.connector
3
4 # Connect to your MySQL database
5 mydb = mysql.connector.connect(
6     host="localhost",
7     user="root",
8     password="",
9     database="muet_test_for_repeater"
10 )
11
12 # Create a cursor object to execute SQL queries
13 mycursor = mydb.cursor()
14
15 # Function to insert data into the table
16 def collect_data():
17     name = name_entry.get()
18     age = age_entry.get()
19     ic = ic_entry.get()
20     semester = semester_type.get()
21
22     package = package_type.get()
23     category = int(category_type.get())
24
25
```

```
26
27 prices = {
28     "Package Smart": 100,
29     "Package Intelligent": 150,
30     "Package Brilliant": 175,
31 }
32
33 total_price = (prices[package]*category)
34 output_label.config(text=f"package: {package}, \ncategory: {category}, \n\nTotal Price: RM{total_price}")
35
36 print("Name:", name)
37 print("Age:", age)
38 print("IC:", ic)
39 print("Semester:", semester)
40
41 # To insert data
42 sql = "INSERT INTO registration (NAME, AGE, IC, SEMESTER, PACKAGE, CATEGORY) VALUES (%s, %s, %s, %s, %s, %s)"
43 val = [(name, age, ic, semester, package, category)]
44 mycursor.execute(sql, val)
45 mydb.commit()
```

```
File Edit Selection View Go Run Terminal Help
registration.py X
registration.py > collect_data
49
50 # Tkinter GUI
51 root = tk.Tk()
52 root.title("MySQL Database with Tkinter")
53 root.geometry("1000x700")
54 root.configure(bg="#FFB90F")
55
56 #Page title=
57
58 label = tk.Label(root, text='MUET TEST FOR REPEATER', font=("calibri", 15, "bold"), fg="blue", bg="orange")
59 label.pack(ipadx=6, ipady=6)
60 #frame=
61
62 frame = tk.Frame(root, bg="#FFB90F")
63 frame.pack()
64
65 #Information about student=
66 #frame(bingo)=
67
68 bingo = tk.LabelFrame(frame, text="", bg="#FFB90F")
69 bingo.grid(row=1, column=1, sticky="news", ipadx=4, ipady=4)
70
71 label_name = tk.Label(bingo, text="Name:", font=("calibri", 14, "bold"), fg="#8B706B", bg="orange")
72 label_name.pack()
73 name_entry = tk.Entry(bingo)
74 name_entry.pack()
75
76 label_age = tk.Label(bingo, text="Age:", font=("calibri", 14, "bold"), fg="#8B706B", bg="orange")
77 label_age.pack()
78 age_entry = tk.Entry(bingo)
79 age_entry.pack()
80
81 label_ic = tk.Label(bingo, text="IC:", font=("calibri", 14, "bold"), fg="#8B706B", bg="orange")
82 label_ic.pack()
83 ic_entry = tk.Entry(bingo)
84 ic_entry.pack()
```

```

File Edit Selection View Go Run Terminal Help
New folder

registration.py X
collect_data
86 label_semester = tk.Label(bingo, text="Semester:", font=("calibri", 14, "bold"), fg="#8B7D6B", bg="orange")
87 label_semester.pack()
88 semester_type = tk.StringVar(bingo)
89 semester_type.set("Select your Semester")
90 semester_dropdown = tk.OptionMenu(bingo, semester_type, 1, 2, 3, 4, 5, 6)
91 semester_dropdown.pack(pady=10)
92
93
94 # Choosing test =
95 # frame(kirby)=
96 kirby = tk.LabelFrame(frame, text="", bg="#FFB90F")
97 kirby.grid(row=0, column=0, sticky="News", ipadx=1, ipady=1)
98
99
100 packs_label = tk.Label(kirby, text="Here are the Test", font=("calibri", 13, "bold"), fg="blue", bg="orange")
101 packs_label.pack(padx=3, pady=10,)
102
103 packs_label = tk.Label(kirby, text="Test A= Listening Test", font=("calibri", 11, ), fg="white", bg="orange")
104 packs_label.pack(padx=2, pady=2)
105
106 packs_label = tk.Label(kirby, text="Test B= Writing Test", font=("calibri", 11, ), fg="white", bg="orange")
107 packs_label.pack(padx=2, pady=2)
108
109 packs_label = tk.Label(kirby, text="Test C= Oral Test", font=("calibri", 11, ), fg="white", bg="orange")
110 packs_label.pack(padx=2, pady=2)
111
112
113
114
115 # Choosing the test based on package
116 # frame(aum)
117
118 aum = tk.LabelFrame(frame, text="", bg="#FFB90F")
119 aum.grid(row=0, column=2, sticky="News", ipadx=1, ipady=1)
120
121
122
123 packs_label = tk.Label(aum, text="Here are the Test based on Package", font=("calibri", 13, "bold"), fg="blue", bg="orange")
124 packs_label.pack(padx=3, pady=10,)
125
126 packs_label = tk.Label(aum, text="Package smart= Test A + Test B", font=("calibri", 11, ), fg="white", bg="orange")
127 packs_label.pack(padx=2, pady=2)
128
129 packs_label = tk.Label(aum, text="Package Intelligent= Test B + Test C", font=("calibri", 11, ), fg="white", bg="orange")
130 packs_label.pack(padx=2, pady=2)
131
132 packs_label = tk.Label(aum, text="Package Brilliant = Test A + Test B + Test C", font=("calibri", 11, ), fg="white", bg="orange")
133 packs_label.pack(padx=2, pady=2)
134
135
136
137
138 lala = tk.LabelFrame(frame, text="", bg="#FFB90F")
139 lala.grid(row=0, column=4, sticky="News", ipadx=1, ipady=1)
140
141
142 packs_label = tk.Label(aum, text="The Prices", font=("calibri", 13, "bold"), fg="blue", bg="orange")
143 packs_label.pack(padx=3, pady=10)
144
145 packs_label = tk.Label(aum, text="Package Smart= RM 100", font=("calibri", 11, ), fg="white", bg="orange")
146 packs_label.pack(padx=2, pady=2)
147
148 packs_label = tk.Label(aum, text="Package Intelligent= RM 150", font=("calibri", 11, ), fg="white", bg="orange")
149 packs_label.pack(padx=2, pady=2)
150
151 packs_label = tk.Label(aum, text="Package Brilliant = RM 175", font=("calibri", 11, ), fg="white", bg="orange")
152 packs_label.pack(padx=2, pady=2)
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

File Edit Selection View Go Run Terminal Help
New folder

registration.py X
collect_data
115 # Choosing the test based on package
116 # frame(aum)
117
118 aum = tk.LabelFrame(frame, text="", bg="#FFB90F")
119 aum.grid(row=0, column=2, sticky="News", ipadx=1, ipady=1)
120
121
122 packs_label = tk.Label(aum, text="Here are the Test based on Package", font=("calibri", 13, "bold"), fg="blue", bg="orange")
123 packs_label.pack(padx=3, pady=10)
124
125 packs_label = tk.Label(aum, text="Package smart= Test A + Test B", font=("calibri", 11, ), fg="white", bg="orange")
126 packs_label.pack(padx=2, pady=2)
127
128 packs_label = tk.Label(aum, text="Package Intelligent= Test B + Test C", font=("calibri", 11, ), fg="white", bg="orange")
129 packs_label.pack(padx=2, pady=2)
130
131 packs_label = tk.Label(aum, text="Package Brilliant = Test A + Test B + Test C", font=("calibri", 11, ), fg="white", bg="orange")
132 packs_label.pack(padx=2, pady=2)
133
134
135
136 lala = tk.LabelFrame(frame, text="", bg="#FFB90F")
137 lala.grid(row=0, column=4, sticky="News", ipadx=1, ipady=1)
138
139
140 packs_label = tk.Label(aum, text="The Prices", font=("calibri", 13, "bold"), fg="blue", bg="orange")
141 packs_label.pack(padx=3, pady=10)
142
143 packs_label = tk.Label(aum, text="Package Smart= RM 100", font=("calibri", 11, ), fg="white", bg="orange")
144 packs_label.pack(padx=2, pady=2)
145
146 packs_label = tk.Label(aum, text="Package Intelligent= RM 150", font=("calibri", 11, ), fg="white", bg="orange")
147 packs_label.pack(padx=2, pady=2)
148
149 packs_label = tk.Label(aum, text="Package Brilliant = RM 175", font=("calibri", 11, ), fg="white", bg="orange")
150 packs_label.pack(padx=2, pady=2)
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

144 packs_label.pack(padx=2, pady=2)
145
146 packs_label = tk.Label(aum, text= "Package Intelligent= RM 150",font= ("calibri", 11, ), fg= "white", bg = "orange")
147 packs_label.pack(padx=2, pady=2)
148
149 packs_label = tk.Label(aum, text= "Package Brilliant = RM 175",font= ("calibri", 11, ), fg= "white", bg = "orange")
150 packs_label.pack(padx=2, pady=2)
151
152
153 #choosing the Category
154 #frame (ish)
155
156 ish= tk.LabelFrame(frame, text="", bg="#FFB90F")
157 ish.grid(row=0, column=1, sticky="News", ipadx= 1, ipady= 1)
158
159
160 packs_label = tk.Label(ish, text= "The Repeater category",font= ("calibri", 13, "bold"), fg= "blue", bg = "orange")
161 packs_label.pack(padx=3, pady=10)
162
163 packs_label = tk.Label(ish, text= "if you are once repeater: please select 1",font= ("calibri", 11, ), fg= "white", bg = "orange")
164 packs_label.pack(padx=2, pady=2)
165
166 packs_label = tk.Label(ish, text= "if you are twice repeater: please select 2",font= ("calibri", 11, ), fg= "white", bg = "orange")
167 packs_label.pack(padx=2, pady=2)
168
169 packs_label = tk.Label(ish, text= "if you are thrice repeater: please select 3",font= ("calibri", 11, ), fg= "white", bg = "orange")
170 packs_label.pack(padx=2, pady=2)
171
172 packs_label = tk.Label(ish, text= "P/S: The price based is multiplication of the price and how many times you repeat",font= ("calibri", 11, ), fg= "white", bg = "orange")
173 packs_label.pack(padx=2, pady=2)
174
175
176 #choosing the option
177 #frame (nono)
178
179
180 nono= tk.LabelFrame(frame, text="", bg="#FFB90F")

```

```

176 #choosing the option
177 #frame (nono)
178
179
180 nono= tk.LabelFrame(frame, text="", bg="#FFB90F")
181 nono.grid(row=1, column=0, sticky="News", ipadx= 1, ipady= 1)
182
183 packs_label = tk.Label(nono, text= "Option", font= ("calibri", 13, "bold"), fg= "blue", bg = "orange")
184 packs_label.pack(padx=3, pady=10,)
185
186 package_type= tk.StringVar(nono)
187 package_type.set("select your package")
188 package_dropdown= tk.OptionMenu(nono, package_type, "Package Smart", "Package Intelligent", "Package Brilliant" )
189 package_dropdown.pack(pady=10)
190
191 category_type= tk.StringVar(nono)
192 category_type.set("select your category")
193 category_dropdown= tk.OptionMenu(nono, category_type, 1, 2, 3 )
194 category_dropdown.pack(pady=10)
195
196
197
198 # Buttons to perform operations
199 # Buttons grid
200
201 save_button = tk.Button(root, text="Done", command= collect_data)
202 save_button.place(x= 800, y = 450, width= 80, height= 50)
203

```

```

204 #prices
205 #frame (nais)
206
207 nais= tk.LabelFrame(frame, text="", bg="#FFB90F")
208 nais.grid(row=1, column=2, sticky="News", ipadx= 1, ipady= 1)
209
210 label= tk.Label(nais,text= "total price", font= ("calibri",12,))
211 label.pack(ipadx=10, ipady=10)
212
213 output_label= tk.Label(nais, text="")
214 output_label.pack(ipadx=15,ipady=10)
215
216
217
218
219 root.mainloop()

```

Snapshot of GUI:

MySQL Database with Tkinter

MUET TEST FOR REPEATER

Here are the Test Test A= Listening Test Test B= Writting Test Test C= Oral Test	The Repeater category if you are once repeater: please select 1 if you are twice repeater: please select 2 if you are thrice repeater: please select 3 P/S: The price based is multiplication of the price and how many times you repeat	Here are the Test based on Package Package smart= Test A + Test B Package Intelligent= Test B + Test C Package Brilliant = Test A + Test B + Test C The Prices Package Smart= RM 100 Package Intelligent= RM 150 Package Brilliant = RM 175
Option Select your package Select your category	Name: Age: Ic: Semester: Select your Semester	total price Done

Snapshot of Database (PHP ADMIN):

The screenshot shows the phpMyAdmin interface with the 'registration' table selected. The table structure is as follows:

NAME	AGE	IC	PACKAGE	SEMESTER	CATEGORY
maizatuli	19	2147483647	Package Smart	3	2
Awatir	19	2147483647	Package Intelligent	3	3
Salsabila	19	2147483647	Package Brilliant	5	1
Aleya	19	2147483647	Package Intelligent	6	3
Sofiyah	19	2147483647	Package Smart	3	1
Fatehah	19	2147483647	Package Brilliant	1	2
Iffah	19	2147483647	Package Intelligent	5	3
Filayah	19	2147483647	Package Smart	3	3
Piqah	19	423456789	Package Brilliant	3	1
jummad	20	2147483647	Package Smart	5	2
Nanad	21	2147483647	Package Brilliant	6	3

The screenshot shows the 'Table structure' view for the 'registration' table. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	NAME	varchar(30)	utf8mb4_general_ci		Yes	NULL			Change Drop More
2	AGE	int(3)			Yes	NULL			Change Drop More
3	IC	int(11)			Yes	NULL			Change Drop More
4	SEMESTER	int(2)			Yes	NULL			Change Drop More
5	PACKAGE	varchar(20)	utf8mb4_general_ci		Yes	NULL			Change Drop More
6	CATEGORY	int(30)			Yes	NULL			Change Drop More

Flowchart:

