



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

Sekolah Pendidikan Profesional dan  
Pendidikan Berterusan  
(UTMSPACE)

**FINAL EXAMINATION / PEPERIKSAAN AKHIR  
SEMESTER I – SESSION 2018 / 2019  
PROGRAM KERJASAMA**

COURSE CODE : DDWC 2623 / DDPC 2623 / DDPC 3143  
*KOD KURSUS*

COURSE NAME : OBJECT - ORIENTED PROGRAMMING USING JAVA  
*NAMA KURSUS PENGATURCARAAN BERORIENTASIKAH OBJEK MENGGUNAKAN JAVA*

YEAR / PROGRAMME : 2 DDWC / 2 DDWZ / 2 DDPC / 2 DDPZ / 3 DDC / 3 DDZ  
*TAHUN / PROGRAM*

DURATION : 2 HOURS AND 30 MINUTES  
*TEMPOH*

DATE : NOVEMBER 2018  
*TARIKH*

**INSTRUCTIONS:**

**ARAHAN**

1. The question paper consists of 3 sections: A, B and C. Answer **ALL** questions in the question paper.  
*Kertas soalan terdiri daripada 3 bahagian: A, B dan C. Jawab **SEMUA** soalan dalam kertas soalan.*
2. Candidates are required to follow all instructions given out by examination invigilators.  
*Calon dikehendaki mematuhi semua arahan daripada penyelia peperiksaan.*

( You are required to write your name and your lecturer's name on your answer script )  
( *Pelajar dikehendaki tuliskan nama dan nama pensyarah pada skrip jawapan* )

NAME / NAMA	:	.....
I.C NO. / NO. K/PENGENALAN	:	.....
YEAR / COURSE TAHUN / KURSUS	:	.....
COLLEGE NAME NAMA KOLEJ	:	.....
LECTURER'S NAME NAMA PENSYARAH	:	.....

This examination paper consists of 13 pages including the cover.  
*Kertas soalan ini mengandungi 13 muka surat termasuk kulit hadapan.*



## PUSAT PROGRAM KERJASAMA

### PETIKAN DARIPADA PERATURAN AKADEMIK ARAHAN AM - PENYELEWENGAN AKADEMIK

#### 1. SALAH LAKU SEMASA PEPERIKSAAN

- 1.1 Pelajar tidak boleh melakukan mana-mana salah laku peperiksaan seperti berikut :-
  - 1.1.1 memberi dan/atau menerima dan/atau memiliki sebarang maklumat dalam bentuk elektronik, bercetak atau apa jua bentuk lain yang tidak dibenarkan semasa berlangsungnya peperiksaan sama ada di dalam atau di luar Dewan Peperiksaan melainkan dengan kebenaran Ketua Pengawas; atau
  - 1.1.2 menggunakan makluman yang diperolehi seperti di atas bagi tujuan menjawab soalan peperiksaan; atau
  - 1.1.3 menipu atau cuba untuk menipu atau berkelakuan mengikut cara yang boleh ditafsirkan sebagai menipu semasa berlangsungnya peperiksaan; atau
  - 1.1.4 lain-lain salah laku yang ditetapkan oleh Universiti (seperti membuat bising, mengganggu pelajar lain, mengganggu Pengawas menjalankan tugasnya).

#### 2. HUKUMAN SALAH LAKU PEPERIKSAAN

- 2.1 Sekiranya pelajar didapati telah melakukan pelanggaran mana-mana peraturan peperiksaan ini, setelah diperakukan oleh Jawatankuasa Peperiksaan Fakulti dan disabitkan kesalahannya, Senat boleh mengambil tindakan dari mana-mana satu yang berikut :-
  - 2.1.1 memberi markah SIFAR (0) bagi keseluruhan keputusan peperiksaan kursus yang berkenaan (termasuk kerja kursus); atau
  - 2.1.2 memberi markah SIFAR (0) bagi semua kursus yang didaftarkan pada semester tersebut.
- 2.2 Jawatankuasa Akademik Fakulti boleh mencadangkan untuk diambil tindakan tatatertib mengikut peruntukan Akta Universiti dan Kolej Universiti, 1971, Kaedah-kaedah Universiti Teknologi Malaysia (Tatatertib Pelajar-pelajar), 1999 bergantung kepada tahap kesalahan yang dilakukan oleh pelajar.
- 2.3 Pelajar yang didapati melakukan kesalahan kali kedua akan diambil tindakan seperti di perkara 2.1.2 dan dicadang untuk diambil tindakan tatatertib mengikut peruntukan Akta Universiti dan Kolej Universiti, 1971, Kaedah-kaedah Universiti Teknologi Malaysia (Tatatertib Pelajar-pelajar), 1999.

**SECTION A [30 Marks] / BAHAGIAN A [30 Markah]**

**Instruction:** Answer all questions with the most suitable answers in the spaces provided.

**Arahan:** Jawab semua soalan dengan jawapan yang paling sesuai pada ruang jawapan yang disediakan.

1. State the basic principle of object-oriented based on the following statements: [6 M]

*Nyatakan prinsip asas berorientasikan-objek berdasarkan pernyataan-pernyataan berikut:*

- a. Focus on outside view of behavior rather than the implementation.

*Fokus pada pandangan luaran kelakuan berbanding dengan perlaksanaan.*

- b. A new class can inherit all the properties of the original class.

*Satu kelas baru boleh mewarisi semua hak milik kelas asal.*

- c. The ability of different objects to respond, each in its own way to identical messages.

*Kebolehan objek-objek yang berbeza untuk bertindak balas kepada mesej yang sama dengan caranya sendiri.*

- d. Enclose data inside an object.

*Menyembunyikan data dalam sesuatu objek.*

**Answer / Jawapan:**

- a. \_\_\_\_\_  
b. \_\_\_\_\_  
c. \_\_\_\_\_  
d. \_\_\_\_\_

2. Name the keyword or operator provided by Java to : [3 M]

*Namakan katakunci atau operator yang disediakan oleh Java untuk:*

- a. Create a new object

*Cipta satu objek baru*

- b. Invoke an object's method

*Panggil kaedah sesuatu objek*

**Answer / Jawapan:**

- a. \_\_\_\_\_  
b. \_\_\_\_\_

3. A private data field cannot be directly accessed by an object. Name a method that can enable [3 M]  
*Medan data persendirian tidak boleh dicapai secara terus oleh objek. Namakan satu kaedah yang membolehkan*
- a private data accessible.  
*data persendirian boleh dicapai.*
  - a private data to be updated.  
*data persendirian boleh dikemaskini.*

**Answer / Jawapan:**

- \_\_\_\_\_
- \_\_\_\_\_

4. State the method that should be used to perform the following tasks: [6 M]

*Nyatakan kaedah yang perlu digunakan untuk menyelesaikan tugas-tugas berikut:*

- To compare the content of two **String** objects.  
*Untuk membandingkan kandungan dua objek String.*
- To convert a character to lowercase in a **String** object.  
*Untuk menukar satu aksara kepada huruf kecil dalam objek String.*
- To check either the specific character is letter or not letter in a **Character** object.  
*Untuk menyemak samada aksara adalah huruf atau bukan huruf dalam objek Character.*
- To delete characters from start index to end index in a **StringBuffer** object.  
*Untuk menghapus aksara-aksara dari indek mula hingga indek akhir dalam objek StringBuffer.*

**Answer / Jawapan:**

- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_
- \_\_\_\_\_

5. Briefly explain the purpose of the following **Object** class's methods: [3 M]

*Terangkan secara ringkas tujuan kaedah-kaedah kelas Object yang berikut:*

**Answer / Jawapan:**

- tostring() method / Kaedah

---

- equals() method / Kaedah

---

6. Explain the difference between method overloading and method overriding. [3 M]

*Terangkan perbezaan antara kaedah 'overloading' dan kaedah 'overriding'.*

**Answer / Jawapan:**

---

---

---

---

7. Determine whether the following statements are true or false: [3 M]

*Tentukan samada pernyataan-pernyataan berikut adalah benar atau palsu:*

- a. A class that contains abstract methods must be abstract.

*Satu kelas yang mengandungi kaedah abstrak mesti abstrak*

- b. An interface can extend one or more interface.

*Satu antaramuka boleh dilanjutkan kepada satu atau lebih antaramuka.*

- c. A subclass cannot be abstract if its superclass is non-abstract.

*Sub kelas tidak boleh menjadi abstrak jika kelas supernya adalah bukan abstrak.*

**Answer / Jawapan:**

- a. \_\_\_\_\_  
b. \_\_\_\_\_  
c. \_\_\_\_\_

8. In an interface, we must define: [3 M]

*Dalam antaramuka, kita mesti takrifkan:*

- a. All data fields are \_\_\_\_\_

*Semua medan data adalah \_\_\_\_\_.*

- b. All methods are \_\_\_\_\_

*Semua kaedah adalah \_\_\_\_\_.*

**Answer / Jawapan:**

- a. \_\_\_\_\_  
b. \_\_\_\_\_

**SECTION B [50 Marks] / BAHAGIAN B [50 Markah]**

**Instruction:** Answer all questions in the spaces provided.

**Arahan:** Jawab semua soalan pada ruang jawapan yang disediakan.

1. Draw a UML class notation representing the following class: The name of the class is Rent. It has three attributes, name, date and telephone number represented by a string value. It also has a method; makeRent(), which returns the object information of string type. Its attributes should be designated as **private** and its method as **public**. [6 M]

*Lukis notasi kelas UML untuk mewakilkan kelas berikut: Nama kelas tersebut ialah Rent. Ianya mempunyai tiga atribut iaitu nama, tarikh dan nombor telefon yang diwakilkan dengan nilai rentetan. Ia juga mempunyai satu kaedah iaitu makeRent() yang mana mengembalikan maklumat objek jenis rentetan. Atributnya perlu ditanda sebagai private dan kaedahnya sebagai public.*

**Answer / Jawapan:**

2. Rewrite the following switch..case statements to the if..else if statements. [5 M]

*Tulis semula penyataan switch..case berikut kepada penyataan if .. else if.*

```
switch(class) {  
    case 'a' : case 'A':  
        status = "business class airlines ticket";  
        break;  
    case 'b': case 'B':  
        status = "economy class airlines ticket"  
        break;  
    default:  
        status = "Invalid code"  
        break;  
}
```

**Answer / Jawapan:**

---

---

---

---

---

---

---

---

---

3. You are given a Salesman class and incomplete TestSalesman class as defined below. In TestSalesman class, write a method that accepts an array of Salesman, instantiates 10 Salesman objects with values of Salesman's id and total sales entered by the user, and then finds the highest salesman's total sales. Display the information of the salesman who obtains the highest total sales. [8 M]

*Anda diberi kelas Salesman dan kelas TestSalesman yang tidak lengkap seperti ditakrifkan di bawah. Dalam kelas TestSalesman, tulis satu kaedah yang menerima satu tatasusunan jenis Salesman, cipta 10 objek Salesman dengan nilai bagi id jurujual dan jumlah jualan dimasukkan oleh pengguna dan kemudian cari jumlah jualan jurujual yang terbanyak. Papar maklumat bagi jurujual yang memperolehi jumlah jualan tertinggi.*

```
public class Salesman {  
    private String salesmanID;  
    private double totalSales;  
    public salesman(String id, double ts) { salesmanID = id;  
                                         totalSales = ts;  
    }  
    public String getSalesmanID() { return salesmanID; }  
    public double getTotalSales() { return totalSales; }  
}
```

### **Answer / Jawapan:**

```
import java.util.Scanner;
public class TestSalesman {
    public static void main (String[] args) {
        Salesman[ ] ob = new Salesman[10];
        findSmallestSales(ob);
    }
    public static void findSmallestSales(Salesman[ ] ob) {
        Scanner s = new Scanner (System.in);
```

}

4. Answer all questions from (a) to (c) based on the following program:

Jawab semua soalan dari (a) hingga (c) berdasarkan aturcara berikut:

```
public class TestString {  
    public static void main(String[] args) {  
        String s1 = new String("Good Luck For Your Final Exam");  
        String s2 = "Good Luck For Your Final Exam";  
        StringBuffer s3 = new StringBuffer("Java is cool");  
        System.out.println("Result is " + (s1 == s2));  
        System.out.println("Result is " + s1.equals(s2));  
        System.out.println("Result is " + s2.indexOf('e'));  
        System.out.println("Result is " + s3.append(" and simple!!"));  
        System.out.println("Result is " + s2.substring(4, 11));  
        System.out.println("Result is " + s3.delete(5, 8));  
    }  
}
```

- a) What is the output of the program above?

[9 M]

*Apakah output bagi aturcara di atas?*

### **Answer / Jawapan:**

---

---

---

---

---

- b) Write a statement to obtain a letter 'F' from s1.

[2 M]

*Tulis satu pernyataan untuk mendapatkan satu huruf 'F' dari s1.*

### **Answer / Jawapan:**

Digitized by srujanika@gmail.com

- c) Write a program fragments to find how many uppercase letter in s2.

[4 M]

*Tulis keratan arucara untuk mencari bilangan huruf besar dalam s2.*

### **Answer / Jawapan:**

---

---

---

---

---

---

---

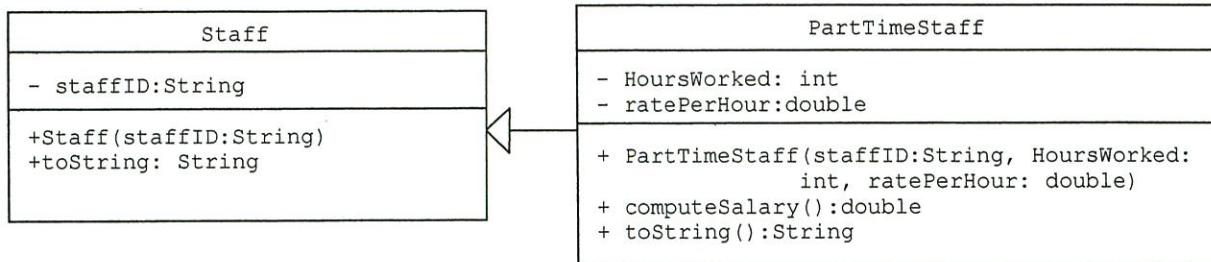
---

---

---

5. Answer questions (a), (b) and (c) based on the following UML class diagram.

Jawab soalan (a), (b) dan (c) berdasarkan rajah kelas UML berikut.



- a) A **Staff** class is a \_\_\_\_\_ while a **PartTimeStaff** class is a \_\_\_\_\_. [2 M]

Kelas Staff ialah \_\_\_\_\_ manakala kelas PartTimeStaff ialah \_\_\_\_\_.

- b) Write Java code for describing PartTimeStaff class includes it's constructor, `toString()` method and `computeSalary()` method. Method `toString()` will return the information of PartTimeStaff that contains `staffID`, `hourWorked`, `ratePerHour` and `salary`. Method `computeSalary()` is used to compute the salary of PartTimeStaff's object by multiplying `hoursWorked` with `ratePerHour`. [8 M]

Tulis kod Java bagi menerangkan kelas PartTimeStaff termasuklah konstruktornya, kaedah `toString()` dan kaedah `computeSalary()`. Kaedah `toString()` akan mengembalikan maklumat PartTimeStaff yang mengandungi `staffID`, `hourWorked`, `ratePerHour` dan gaji. Kaedah `computeSalary()` digunakan untuk mengira gaji objek PartTimeStaff dengan mendarabkan `hoursWorked` dengan `ratePerHour`.

### **Answer / Jawapan:**

c) Write a test program to do the following tasks:

[6 M]

*Tulis satu aturcara ujian untuk melakukan tugas-tugas berikut:*

- i. Create one PartTimeStaff object named myPartTimeStaff with appropriate values for its attributes.  
*Cipta satu objek PartTimeStaff yang bernama myPartTimeStaff dengan nilai-nilai yang sesuai bagi atribut-atributnya.*
  - ii. Display the information of myPartTimeStaff object via a message dialog box.  
*Papar maklumat objek myPartTimeStaff melalui kotak dialog mesej.*

**Answer / Jawapan:**

**SECTION C [20 Marks] / BAHAGIAN C [20 Markah]**

**Instruction:** Write a complete program based on the question requirement given.

**Arahan:** Tulis aturcara lengkap berdasarkan keperluan soalan yang diberikan.

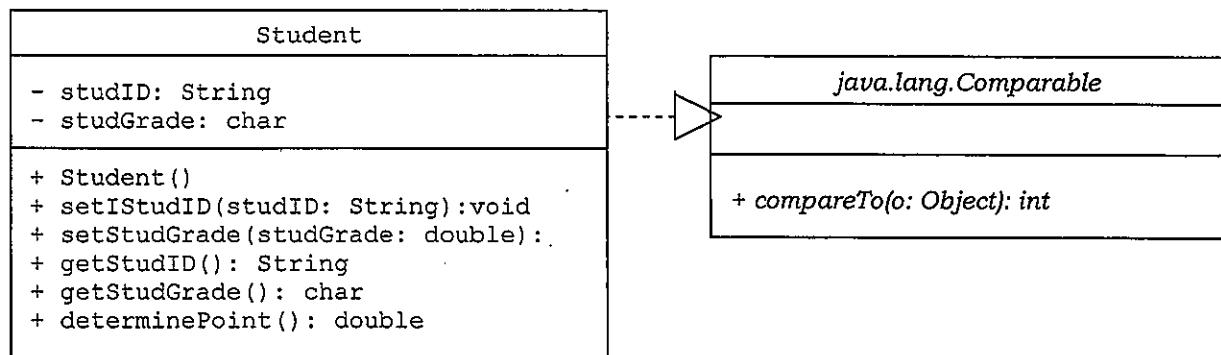


Figure 1: Shows the relationship between class `Student` and interface `Comparable`.

Rajah 1: Menunjukkan perhubungan antara kelas `Student` dan antaramuka `Comparable`.

Write a program for a class `Student` based on the Figure 1. The `Student` class has two attributes, one constructor, two mutator methods, two accessor methods and `determinePoint()` methods. The `determinePoint()` method will return a point obtained based on the grade. The grade and its associated point can be referred on a table below. The `Student` class implements `compareTo()` method from `java.lang.Comparable` interface to compare the point obtained for two objects. Write a test program that creates two `Student` objects with values of attributes are entered by the user. Display the information of these two objects includes their id, grade, point and also the point comparison result.

Tulis aturcara bagi kelas `Student` berdasarkan Rajah 1. Kelas `Student` mempunyai dua atribut, satu konstruktor, dua kaedah 'mutator', dua kaedah 'accessor' dan kaedah `determinePoint()`. Kaedah `determinePoint()` akan memulangkan mata nilai yang diperolehi berdasarkan kepada gred. Gred dan mata nilai berkaitan boleh dirujuk pada jadual di bawah. Kelas `Student` melaksanakan kaedah `compareTo()` dari antaramuka `java.lang.Comparable` untuk membandingkan mata nilai yang diperolehi bagi dua objek. Tulis satu aturcara ujian yang mencipta dua objek `Student` dengan nilai-nilai atribut yang dimasukkan oleh pengguna. Papar maklumat kedua objek ini termasuklah id, gred, mata nilai dan juga hasil perbandingan mata nilai.

Grade / Gred	Point / Mata Nilai
A	4.00
B	3.00
C	2.00
D	1.00

○

### **Answer / Jawapan:**

### **Answer / Jawapan:**

**END OF QUESTIONS**

**Mukasurat ini sengaja dibiarkan kosong**

*[ This page is purposely left blank ]*