



FINAL EXAMINATION / PEPERIKSAAN AKHIR
SEMESTER II – SESSION 2020 / 2021
PROGRAM KERJASAMA

COURSE CODE : DDWC 2623
KOD KURSUS

COURSE NAME : OBJECT – ORIENTED PROGRAMMING USING JAVA
NAMA KURSUS : *PENGATURCARAAN BERORIENTASIKAN OBJEK MENGGUNAKAN JAVA*

YEAR / PROGRAMME : 2 DDWC
TAHUN / PROGRAM

DURATION : 3 HOURS (INCLUDING SUBMISSION HOUR)
TEMPOH : *3 JAM (TERMASUK MASA PENGHANTARAN)*

DATE : APRIL / MAY 2021
TARIKH

INSTRUCTION / ARAHAN:

1. The question paper consists of **3 PARTS**: A, B and C.
*Kertas soalan terdiri daripada **3 BAHAGIAN**: A, B dan C.*
2. Answer **ALL** questions and write your answers on the answer sheet.
*Jawab **SEMUA** soalan dan tulis jawapan anda pada kertas jawapan.*
3. Write your name, matric no., identity card no., course code, course name, section no. and lecturer's name on the first page (in the upper left corner) and every page thereafter on the answer sheet.
Tulis nama anda, no. matrik, no. kad pengenalan, kod kursus, nama kursus, no. seksyen dan nama pensyarah pada muka surat pertama (penjuru kiri atas) kertas jawapan dan pada setiap muka surat jawapan.
4. Each answer sheet must have a page number written at the bottom right corner.
Setiap helai kertas jawapan mesti ditulis nombor muka surat pada bahagian bawah penjuru kanan.
5. Answers should be handwritten, neat and clear.
Jawapan hendaklah ditulis tangan, kemas dan jelas menggunakan huruf cerai.

WARNING / AMARAN

Students caught copying / cheating during the examination will be liable for disciplinary actions and the faculty may recommend the student to be expelled from sitting for exam.
Pelajar yang ditangkap meniru / menipu semasa peperiksaan akan dikenakan tindakan disiplin dan pihak fakulti boleh mengesyorkan pelajar diusir dari menduduki peperiksaan.

This examination paper consists of **8** pages including the cover.
*Kertas soalan ini mengandungi **8** muka surat termasuk kulit hadapan.*

ONLINE EXAMINATION RULES AND REGULATIONS
PERATURAN PEPERIKSAAN SECARA DALAM TALIAN

1. Student must carefully listen and follow instructions provided by invigilator.
Pelajar mesti mendengar dan mengikuti arahan yang diberikan oleh pengawas peperiksaan dengan teliti.
2. Student is allowed to start examination only after confirmation of invigilator if all needed conditions are implemented.
Pelajar dibenarkan memulakan peperiksaan hanya setelah pengesahan pengawas peperiksaan sekiranya semua syarat yang diperlukan telah dilaksanakan.
3. During all examination session student has to ensure, that he is alone in the room.
Semasa semua sesi peperiksaan pelajar harus memastikan bahawa dia bersendirian di dalam bilik.
4. During all examination session student is not allowed to use any other devices, applications except other sites permitted by course lecturer.
Sepanjang sesi peperiksaan pelajar tidak dibenarkan menggunakan peranti dan aplikasi lain kecuali yang dibenarkan oleh pensyarah kursus.
5. After completing the exam student must inform invigilator via the set communication platform (eg. WhatsApp etc.) about completion of exam and after invigilator's confirmation leave examination session.
Selepas peperiksaan selesai, pelajar mesti memaklumkan kepada pengawas peperiksaan melalui platform komunikasi yang ditetapkan (contoh: Whatsapp dan lain-lain) mengenai peperiksaan yang telah selesai dan meninggalkan sesi peperiksaan selepas mendapat pengesahan daripada pengawas peperiksaan.
6. Any technical issues in submitting answers online have to be informed to respective lecturer within the given 30 minutes. Request for re-examination or appeal will not be entertain if complains are not made by students to their lecturers within the given 30 minutes.
Sebarang masalah teknikal dalam menghantar jawapan secara dalam talian perlu dimaklumkan kepada pensyarah masing-masing dalam masa 30 minit yang diberikan. Permintaan untuk pemeriksaan semula atau rayuan tidak akan dilayan sekiranya aduan tidak dibuat oleh pelajar kepada pensyarah mereka dalam masa 30 minit yang diberikan.
7. During online examination, the integrity and honesty of the student is also tested. At any circumstances student is not allowed to cheat during examination session. If any kind of cheating behaviour is observed, UTM have a right to follow related terms and provisions stated in the respective Academic Regulations and apply needed measures.
Semasa peperiksaan dalam talian, integriti dan kejujuran pelajar juga diuji. Walau apa pun keadaan pelajar tidak dibenarkan menipu semasa sesi peperiksaan. Sekiranya terdapat sebarang salah laku, UTM berhak untuk mengikuti terma yang dinyatakan dalam Peraturan Akademik.

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

INSTRUCTION : ANSWER ALL QUESTIONS.

ARAHAN : JAWAB SEMUA SOALAN.

1. State and explain **THREE (3)** advantages of object-oriented approach. Give an example for each of advantages in Java program. **[6 M]**

Nyatakan dan jelaskan TIGA (3) kebaikan pendekatan berorientasikan objek. Berikan satu contoh bagi setiap kebaikan tersebut dalam program Java.

2. Write Java syntax to perform the following tasks: **[6 M]**

Tulis sintak Java untuk menjalankan tugas-tugas berikut:

- a. To retrieve a substring in a **String** object .

Untuk mencapai satu sub rentetan dalam objek String.

- b. To replace an original character with a new character in a **String** object.

Untuk menggantikan satu aksara asal kepada satu aksara baharu dalam objek String.

- c. To check either the specific character is digit in a **Character** object.

Untuk menyemak samada aksara adalah digit dalam objek Character.

- d. To insert a new message in a **StringBuffer** object.

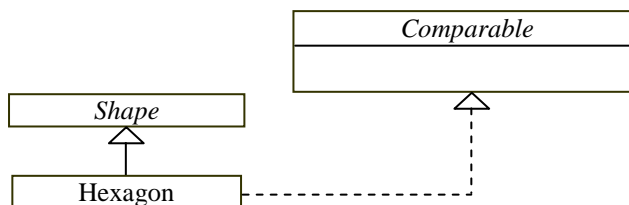
Untuk selit satu mesej baru dalam objek StringBuffer.

3. Briefly explain the **THREE (3)** similarities between abstract class and interface. **[6 M]**

Jelaskan secara ringkas TIGA (3) persamaan antara kelas abstrak dan antaramuka.

4. Answer question (a) and (b) based on the UML class diagram below,

Jawab soalan (a) dan (b) berdasarkan rajah kelas UML di bawah,



- a) Identify abstract class, interface and subclass. **[3 M]**

Kenal pasti kelas abstrak, antaramuka dan kelas sub.

- b) Write Java code to define Shape, Hexagon and Comparable. **[3 M]**

Tulis kod Java untuk menakrifkan Shape, Hexagon dan Comparable.

5. Describe the purpose of the following keywords in Java program: **[6 M]**

Terangkan tujuan kata kunci yang berikut dalam program Java:

a) instanceof

b) implementations

c) super

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

INSTRUCTION : ANSWER ALL QUESTIONS.

ARAHAN : JAWAB SEMUA SOALAN.

1. Draw a **Class Diagram** to develop College Management System for Bestari College that consists of all classes, relationships among classes and multiplicity based on the requirement specification as follows: "A college has one or more departments. Each department has at least ten lecturers. Every student may register any number of courses and every course may have maximum forty students. For every course there is at least one lecturer. Each lecturer may teach any number of courses". **[7 M]**

Lukis **Gambarajah Kelas** untuk membangunkan Sistem Pengurusan Kolej bagi Kolej Bestari yang mengandungi semua kelas, perhubungan antara kelas dan 'multiplicity' berdasarkan spesifikasi keperluan seperti berikut: "Kolej mempunyai satu atau lebih jabatan. Setiap jabatan mempunyai sekurang - kurangnya sepuluh pensyarah. Setiap pelajar boleh mendaftar sebarang bilangan kursus dan setiap kursus boleh mempunyai maksimum empat puluh pelajar. Bagi setiap kursus terdapat sekurangnya seorang pensyarah mengajar. Setiap pensyarah boleh mengajar sebarang bilangan kursus".

2. a) Write a Java program fragment to display dialog box as shown as below: **[4 M]**

Tuliskan satu keratan program Java untuk memaparkan kotak dialog seperti ditunjukkan di bawah:



- b) You are given a method header as below. Write the implementation of this method to find and print the highest and lowest score from an array named `score`. **[5 M]**

Anda diberikan satu pengepala kaedah seperti di bawah. Tulis pelaksanaan bagi kaedah ini untuk mencari dan mencetak markah tertinggi dan terendah dari satu tatasusunan bernama `score`.

```
public void findScore(double[] score)
```

3. You are given a `Book` class as below. Prepare a test program for this class to do the following tasks:

Anda diberikan satu kelas `Book` seperti berikut. Sediakan satu program ujian bagi kelas ini untuk melakukan tugas yang berikut:

Book
+ title: String + price: double + quantity: int
+ Book() + totalPriceofBookPurchased(): double

- a) Create an array of twenty `Book` objects using default constructor. **[2 M]**
Cipta satu tatasusunan bagi dua puluh objek `Book` menggunakan konstruktor lalai.
- b) Set `title`, `price` and `quantity` for each object based on the user's input entered via console. **[6 M]**
Setkan `title`, `price` dan `quantity` bagi setiap objek berdasarkan kemasukan pengguna melalui konsol.
- c) Display all information of `Book` including the total price of book purchased for each object by using dialog box. **[3 M]**
Paparkan semua maklumat `Book` termasuklah jumlah harga buku yang telah dibeli bagi setiap objek menggunakan kotak dialog.

4. Answer all questions (a) to (d) based on the program fragment given as follow:

Jawab semua soalan (a) hingga (d) berdasarkan keratan program diberikan seperti berikut:

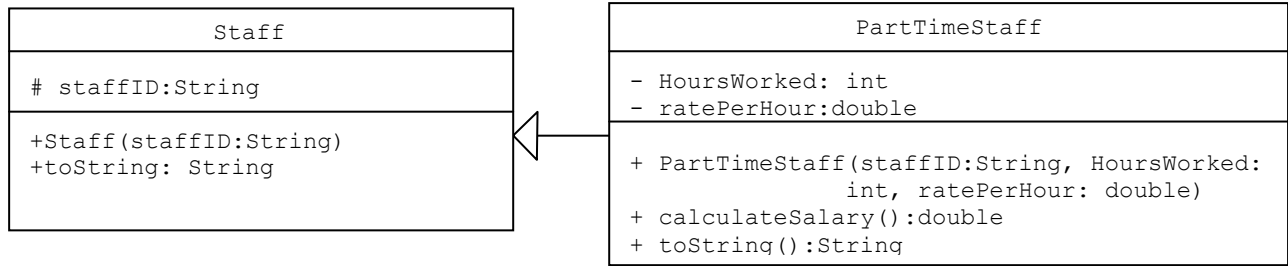
```
StringBuffer s1 = new StringBuffer("Java 001 is robust language 2021.");  
String s2 = "Java is secure programming language."  
int count = findNoOccurence(s2, 'e');  
int result = findDigit(s1);
```

- a) Write Java statement to extract a word "gram" from `s2`. **[2 M]**
Tuliskan pernyataan Java untuk mendapatkan perkataan "gram" dari `s2`.
- b) Write Java statement to delete a word "ust " from `s1`. **[2 M]**
Tulis pernyataan Java untuk menghapuskan perkataan "ust " dari `s1`.
- c) Write a body implementation for `findDigit()` method to find and return the number of digit in `s1`. **[4 M]**
Tulis badan pelaksanaan bagi kaedah `findDigit()` untuk mencari dan mengembalikan bilangan digit dalam `s1`.
- d) Write a method that finds and return the number of occurrence of 'e' letter in the `s2` using the following header: **[4 M]**
Tulis satu kaedah yang mencari dan memulangkan bilangan kewujudan huruf 'e' dalam `s2` menggunakan pengepala kaedah berikut:

```
public static int findNoOccurence(String msg2, char letter)
```

5. Write Java statements to describe `PartTimeStaff` class based on the UML class diagram and given requirements as follow:

Tulis pernyataan Java untuk menerangkan kelas `PartTimeStaff` berdasarkan rajah kelas UML dan keperluan yang diberikan seperti yang berikut:



- a) Define `PartTimeStaff` class / *Takrif kelas `PartTimeStaff`.* **[2 M]**
- b) Write a parameterized constructor for a `PartTimeStaff` class that accepts three arguments for each its data fields. **[3 M]**
Tulis satu konstruktor berparameter bagi kelas `PartTimeStaff` yang menerima tiga argumen bagi setiap medan datanya.
- c) Write an implementation for `calculateSalary()` method to compute the salary of `PartTimeStaff`'s object by multiplying `hoursWorked` with `ratePerHour`. **[3 M]**
Tulis pelaksanaan bagi kaedah `calculateSalary()` untuk mengira gaji objek `PartTimeStaff` dengan mendarabkan `hoursWorked` dengan `ratePerHour`.
- d) Write an implementation for `toString()` method in the `PartTimeStaff` class that overrides `toString()` defined in the `Staff` class. This method will return the information of `PartTimeStaff` that contains `staffID`, `hourWorked`, `ratePerHour` and salary. **[3 M]**
Tulis pelaksanaan bagi kaedah `toString()` dalam kelas `PartTimeStaff` yang 'override' `toString()` yang ditakrifkan dalam kelas `Staff`. Kaedah ini mengembalikan maklumat `PartTimeStaff` yang mengandungi `staffID`, `hourWorked`, `ratePerHour` dan gaji.

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

INSTRUCTION: WRITE A COMPLETE JAVA PROGRAM.

ARAHAN: TULIS PROGRAM JAVA YANG LENGKAP.

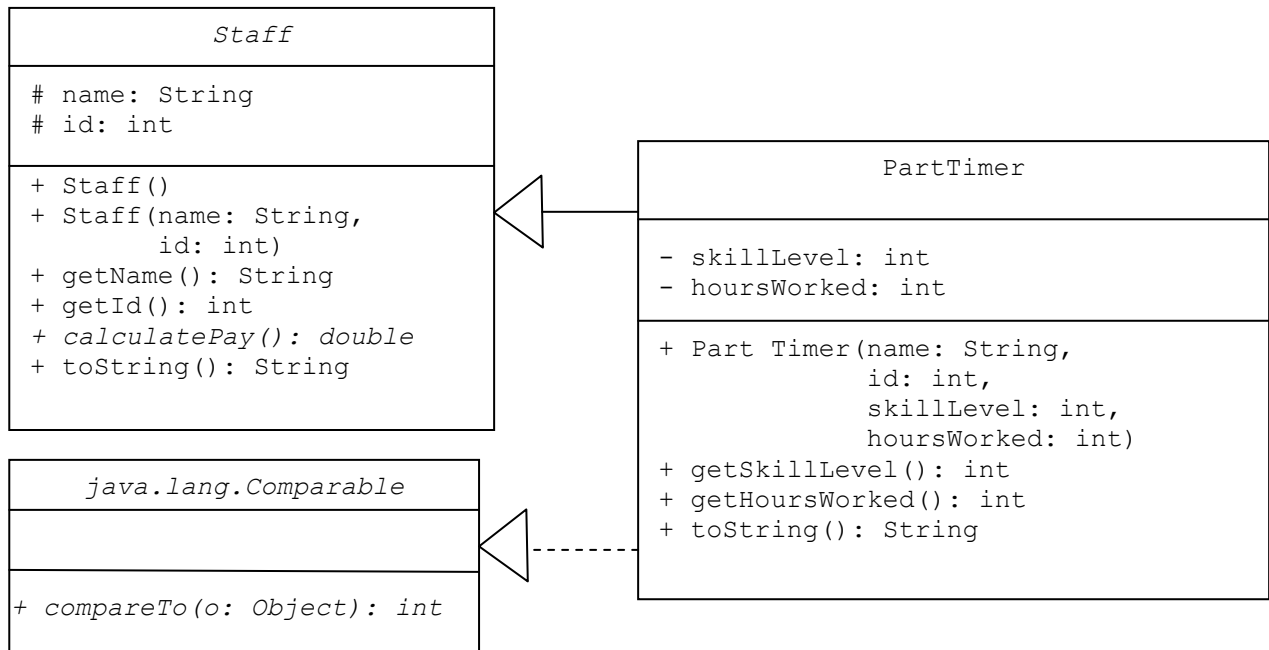


Figure 1: The relationship between Staff, PartTimer and Comparable

Rajah 1: Perhubungan antara Staff, PartTimer dan Comparable

Skill Level / Tahap Kemahiran	Pay Rate Per Hour / Kadar Bayaran Per Jam
0 (Basic / Asas)	RM50.00
1 (Moderate / Sederhana)	RM100.00
2 (Expert / Mahir)	RM150.00

Table 1: Skill level and the pay rate per hour

Jadual 1: Tahap kemahiran dan kadar bayaran per jam

Write a complete Java program to create `PartTimer` class and test class based on the Figure 1, Table 1 and the following requirements: `PartTimer` class is derived from `Staff` and implements `Comparable` Interface. The `calculatePay()` method from the `Staff` class will be implemented in the subclass to calculate the employee's pay. The pay for `PartTimer` is based on the skill level of staff as shown in Table 1, which is the payment can be calculated by multiplying hours worked with the pay rate per hour. The method `toString()` will return the information of `PartTimer` object. The `PartTimer` class implements `compareTo()` method from `java.lang.Comparable` interface to compare the payment of two objects. Then write a test program that creates two `PartTimer` objects using parameterized constructor with any values assigned to their data fields. Then display all the information includes the comparison result of the payment for these two objects.

Tulis satu program Java yang lengkap bagi mencipta kelas `PartTimer` dan kelas ujian berdasarkan Rajah 1, Jadual 1 dan keperluan yang berikut: Kelas `PartTimer` dihasilkan daripada kelas `Staff` dan melaksanakan antaramuka `Comparable`. Kaedah `calculatePay()` daripada kelas `Staff` akan dilaksanakan dalam kelas sub untuk mengira bayaran staf. Bayaran kepada `PartTimer` adalah berdasarkan tahap kemahiran pekerja seperti ditunjukkan dalam Jadual 1, di mana pembayaran boleh dikira dengan mendarabkan jam bekerja dengan kadar bayaran per sejam. Kaedah `toString()` akan mengembalikan maklumat objek `PartTimer`. Kelas `PartTimer` melaksanakan kaedah `compareTo()` dari antaramuka `java.lang.Comparable` untuk membandingkan pembayaran bagi dua objek. Kemudian tulis aturcara ujian yang mencipta dua objek `PartTimer` menggunakan konstruktor berparameter dengan sebarang nilai yang diumpukkan kepada medan data. Kemudian papar semua maklumat termasuk hasil perbandingan pembayaran bagi kedua objek ini.

END OF QUESTIONS / SOALAN TAMAT