

**It is important to note that this Practice Exam Paper is intended only to show the following :**

- the *general format* of the final exam paper
- an *indication* of the type of questions that you can expect in the final exam paper

No solutions will be provided for the sample exam. Your tutor is not responsible for checking your answers for you - check your own answers by referring to the lecture notes (for the theory questions) and typing in the code in BlueJ (for the practical questions).

The actual final exam paper may not have the exact same number of questions as the sample exam paper. The mark allocations for each section may also vary.

**\*\*\*\* Do not memorise the sample questions \*\*\*\***

It is strongly recommended that you visit the "[How eExam Works](#)" website, which contains a lot of useful information about eExams. In particular, there is a [General Knowledge Practice Exam](#) that incorporates all the eExam question types, for you to practice on answering questions on that platform.

EXAM CODE: FIT9131  
Semester 2 2021  
TITLE OF PAPER: Mock Exam 1  
EXAM DURATION: 2 hours 10 mins

## Instructions

**Answer all questions on the exam.**

Please note:

- You are not permitted to access any other resources (applications, websites, textbooks) outside of the exam environment.
- The use of online dictionaries or translators are not permitted.
- The use of BlueJ (or other Java development environment) is not permitted.
- For the coding questions, you need to write and format your code clearly, and follow the FIT9131 Java Coding Standards.

An operator precedence table has been provided in case you need this information.

## Authorised Materials: None

### Closed Book

This is a **closed book** assessment. You're not permitted to use any notes, texts, websites or other reference material to assist you in answering the questions.

## Rules

During an exam, you must not have in your possession any item/material that has not been authorised for your exam. This includes books, notes, paper, electronic device/s, mobile phone, smart watch/device, calculator, pencil case, or writing on any part of your body. Any authorised items are listed above. Items/materials on your desk, chair, in your clothing or otherwise on your person will be deemed to be in your possession.

You must not retain, copy, memorise or note down any exam content for personal use or to share with any other person by any means following your exam.

You must comply with any instructions given to you by an exam supervisor.

As a student, and under Monash University's Student Academic Integrity procedure, you must undertake your in-semester tasks, and end-of-semester tasks, including exams, with honesty and integrity. In exams, you must not allow anyone else to do work for you and you must not do any work for others. You must not contact, or attempt to contact, another person in an attempt to gain unfair advantage during your exam session. Assessors may take reasonable steps to check that your work displays the expected standards of academic integrity.

Failure to comply with the above instructions, or attempting to cheat or cheating in an exam may constitute a breach of instructions under regulation 23 of the Monash University (Academic Board) Regulations or may constitute an act of academic misconduct under Part 7 of the Monash University (Council) Regulations.

## Information

Answer all questions on the exam.

### Operator Precedence Table

The following table shows the order of precedence and associativity of operators. Operators higher in the list have higher precedence. Operators in the two top groups have the same order of precedence

OPERATOR	ASSOCIATIVITY
() [] .	left to right
++ --	right to left <sup>a</sup>
++ --	left to right <sup>b</sup>
+ - !	right to left
new (cast)	right to left
* / %	left to right
+ -	left to right
< <= > >=	left to right
== !=	left to right
&&	left to right
	left to right
? :	right to left
= += -= *= /= %=	right to left

<sup>a</sup> post increment, post decrement operators <sup>b</sup>

pre increment, post decrement operators

## Section A - Multiple-Choice Questions (2x10 = 20 Marks)

### Question 1

A program has an integer instance variable called **length**. One of its methods has an integer parameter called **length**. When the name **length** is used within that method:

Select one:

- a. it generates a syntax error
- b. it generates a runtime error
- c. it refers to the instance variable
- d. it refers to the parameter

### Question 2

Which of the following statements is/are **true** (pick only the best answer)?

Select one:

- a. scope of a variable specifies where the variable can be accessed
- b. access modifiers specify which fields and methods within an object are accessible to other objects
- c. using access modifiers facilitates information hiding
- d. all of (a)-(c) are correct
- e. none of (a)-(c) are correct

### Question 3

In Java, the keyword **NULL** refers to:

Select one:

- a. an object with uninitialized attributes
- b. a reference to nothing
- c. an empty string
- d. all of (a)-(c)

**Question 4-10: not shown**

## Section B – Coding / Code Interpretation (25 Marks)

### Question 11

After the following declaration:

```
People[ ] crowd = new People[10];
```

how many **People** objects are there? Explain your answer.

(5 marks)

### Question 12

Write a declaration for a variable named **tutorialGroup** that can hold an **ArrayList** of objects of the class **Student**. Then write a Java statement to create the actual **ArrayList**.

(5 marks)

### Question 13

State the **overall purpose** of the following segment of code. Assume that **a**, **b**, and **c** are declared to be of type **int** and have been initialized. Note that we are looking for the overall goal of the code, not an explanation of each individual line. **The answer must be a single short sentence.**

```
c = a;
```

```
a = b;
```

```
b = c;
```

(5 marks)

**Question 14-15: not shown**

## **Section C - Design/ Testing (20 Marks)**

### **Question 16**

What is the difference between *testing* and *debugging*?

(4 marks)

### **Question 17**

This semester we discussed *modularisation* as a good programming design technique. Explain what *modularisation* means. Include in your answer an example of how you used this in your assignments (just choose one class to describe).

(4 marks)

**Question 18-20: not shown**

## Section D – Coding / Debugging / Assignment-related (35 Marks)

### Question 21

A programmer has written code for a machine that dispenses items. Part of the program should allow the user to pay for their selected item. The user should keep inserting coins until they have paid for their purchase. Once the price of the item has been inserted the program should display a message asking the user to collect their item and, if applicable, also show the correct change owed to them.

If the user cancels their purchase at any time up until they have paid the price of the item, they will be asked to collect the correct amount of money they have inserted as change. Any invalid input should cause an error message to be displayed to the user.

The complete program compiles and runs, but the program does not behave as expected. There are several logic errors in the program. **Find and fix any 4 (four) of these errors.**

When answering this question, please state the line number which you think is incorrect **AND** suggest the correct code which needs to be used instead.

**Only select 4 errors, no marks will be awarded for extra errors found.**

```
1. public void dispenseItem(double price)
2. {
3.     double amount = 0;
4.     char choice = ' ';
5.     Scanner console = new Scanner(System.in);
6.     do
7.     {
8.         System.out.println("A. $1.00");
9.         System.out.println("B. $0.50");
10.        System.out.println("C. $0.20");
11.        System.out.println("D. $0.10");
12.        System.out.println("Z. Cancel");
13.        System.out.print("Please select: ");
14.        choice = console.nextLine().charAt(0);
15.        switch(choice)
16.        {
17.            case 'A': amount += 1.00; break;
18.            case 'B': amount += 0.50; break;
19.            case 'C': amount += 0.20; break;
20.            case 'D': amount += 0.10; break;
21.            case 'Z':
22.                default: System.out.print("Invalid – Press enter to continue");
23.                console.nextLine();
24.        }
25.    } while (choice != 'Z' || amount <= price);
26.
27.    if (choice == 'Z')
28.        System.out.println("Item ready!");
29.    else
30.        System.out.println("Item cancelled!");
31.    if ((choice == 'Z' && amount > 0) || (choice != 'Z' && amount > price))
32.    {
33.        if (choice != 'Z')
34.            amount -= amount;
35.        System.out.println("Your change is $" + price);
36.    }
37. }
```

(1.5x4 = 6 marks)



## Question 22

The following code uses a **for** loop to process an array of integers:

```
int count = 0;
for (int index = 0; index < numbers.length; index++)
    if (numbers[index] % 3 == 0)
        count++;
System.out.println(count);
```

Rewrite the code using a **while** loop. Your code must perform exactly the same function as the given code above. Assume that all variables shown have been declared and initialised. You must not declare any new variable.

(6 marks)

## Question 23-24: not shown

These 2 questions will be **related to your assignments**, e.g. array/ArrayList manipulations, object interactions, File I/O, Exception-handling, data-processing via loops, data validations, etc.

(23 marks)