Sri Lanka Institute of Information Technology

B. Sc. Honours Degree
in
Information Technology

Final Examination
Year 2, Semester I (2018)

IT2030 – Object Oriented Programming
Paper 1a

Duration: 3 Hours

May/June 2018

Online Exam

Instructions to Candidates:

- This paper contains **Four** questions. **Answer All** Questions.
- Marks for each question are given in the paper.
- Total Marks: 100.
- Create a separate Project for each question. The name of the project is provided. Save each Java program using the class name given.
- Store all your program files in the Desktop Folder provided
- This paper contains **9** pages with the Cover Page.

Question 1 (20 marks)

a) Implement an **abstract class** called `Item` to do the following.

   i) It should store the following **properties**
      `itemNo, description, unitPrice`
   ii) Implement a **constructor** to get the three properties as parameters and initialize them.
   iii) Implement a `Display()` **method** which displays the three properties.

Marks 5

b) Implement a **sub class** called `Book` that **extends** the `Item` abstract class.

   i) Add the following **properties**
      `publisher, category, pages`
   ii) Implement a **constructor** to get all the six properties of this class as parameters and initialize them.
   iii) Override the `Display()` **method** to display details of all the properties.

Marks 5

c) Implement a **sub class** called `Car` that extends the `Item` abstract class.

   i) Add the following properties
      `model, type`
   ii) Implement a **constructor** to get all the five properties of this class as parameters and initialize them.
   iii) Override the `Display()` **method** to display details of all the properties.

Marks 5

d) Implement a class called `MainApp` that contains the `main()` function.

   i) Create an `ArrayList` of the `Item` class. Use generics so that only the `Item` and its descendent classes can be stored in the `ArrayList`.
   ii) Create two `Book` type objects and store them in the `ArrayList`.
   iii) Create two `Car` type objects and store them in the `ArrayList`.
   iv) Using a for each loop call the `Display()` method of the books and cars stored in the `ArrayList`

Marks 5

Save the Project as **Ques01**.

Question 2 (25 marks)

a) In this question, you will develop a program that has two **threads** performing a calculation after a countdown.

i) Implement a **Thread** called `CountDown`. Extend the **Thread** class to create it. It should display numbers 1 to 10. Each number should be printed after a one second interval.

Hint : use the `sleep()` method

Marks 4

ii) Implement a **Thread** called `CalcSum`. This class should **implement** the `Runnable` interface.
The thread should calculate the sum of the numbers 1 to 100,000 and display the result on the screen after the calculation is completed. The name of the current thread should also be displayed when the thread is printed.

Hint : The `currentThread()` static method returns the current thread. The `getName()` method returns the name of the thread.

Marks 4

iii) Create a **class** called `MainThreadApp` that contains the `main()` function. In the main function do the following

1) Create a thread object of the `CountDown` class.
2) Create two threads of the `CalcSum` class. Name the two threads as "Black" and "White"
3) Start the `CountDown` thread.
4) The two `CalcSum` threads should start only after the `CountDown` thread has finished execution.

Hint : use the `setName()` method to name a thread.

Marks 4

Save the Project as **Ques02a**.

b) In this question, you will use **threads** to compute the factorial of a large number.

i) Implement a **class** called `Calculation`.

1) Have a `double` type property called `ans`
2) Implement a **getter** for `ans`
3) Implement a method called
`void Factorial(int start, int end)`
which should compute the multiplication of the numbers between `start` and `end` and store it in `ans`.
e.g. `Factorial(5,10)` → `5x6x7x8x9x10`
This method should be implemented in such a way that multiple threads should be able to access it.

Marks 4

ii) Implement a **Thread** called `ParallelThread`

1) Have the following properties
`myCalc` (`Calculation` type), `start`, `end` (integer type)
2) Implement a **constructor** to get the three properties as parameters and initialize them.
3) The **thread** should compute the factorial for values between the range `start` and `end`.

Marks 4

iii) Implement a **class** called `MainThreadApp` that contains the `main()` function. It should do the following.

1) Create an object of the `Calculation` class.
2) Create four threads of the `ParallelThread` class making use of the `Calculation` object created earlier.
3) Use the four threads to compute the factorial of 40 by dividing the work equally among the four threads.
4) Finally display the factorial of 40

Marks 5

Save the Project as **Ques02b**.

Question 3 (25 marks)

a) A program is required to process students marks in an examination. Implement the following classes that makes use of **exception handling**.

i) Implement a **user defined exception class** called `MarksException`.

1) Have a property called `marks`
2) Implement a **constructor** to get the `marks` property as a parameter and initialize it
3) Implement a **getter** for the `marks` property.

Marks 4

ii) Implement a **class** called `Student`

1) Have the following **properties**
`id, names, marks[]` and `noOfSubjects`
(id and noOfSubjects are integers, marks is a float array)

Marks 2

2) Implement a **constructor** to get values for the properties `id`, and `name` as parameters and initialize them.

Marks 2

3) Implement a **method** called `float inputMarks(int index)` which allows you enter one mark from the keyboard and return it. Here `index` is the subject number of the marks.

Note : index is an integer which is greater than zero.

1. If the marks < 0 or marks > 100 throw a `MarksException`
2. Ignore any errors that can occur through keyboard Input.

Marks 6

4) Implement a **method** called `void input()` which allows you to enter all the marks of a student.

1. Input a value for `noOfSubjects`
2. Input values for the `marks` using the `inputMarks()` method
3. Handle `MarksException` and possible errors when entering the input value for `noOfSubjects`

Marks 5

5) Implement a **method** called `float getAverage()` to calculate the average of the marks stored in the `marks[]` array. Handle the **Division by zero** error which can happen if there are no marks entered.

Marks 3

iii) Implement a **class** called `MainApp` which has a `main()` function.

1) Create a `student` object.
2) Call the `input()` method
3) Display the average using the `getAverage()` method.

Marks 3

Save the Project as **Ques03**.

Question 4 (30 marks)

a) A credit card validation **class** called `CreditCard` needs to be implemented as a **Singleton**.

i) Implement the `CreditCard` class as a `Singleton`.
ii) Implement the method called
`bool validate(String cardno, int code)`
The length of the `cardno` should be 16 and the code should be divisible by the number three.
iii) Create a class called `MainApp` that has the `main()` function.

1) Create two objects of the `CreditCard` class
2) Call the `validate()` method to check if the credit card validation works.
3) Verify that both objects refer to a single `CreditCard` object.

Marks 7

Save the Project as **Ques04a**.

b) A Mobile Phone company has partnered with a construction firm to build houses with automation built in. A software designer has suggested that the **Command Design pattern** can be used for this purpose.

i) Create a **Receiver class** called `Oven`. The `Oven` class should have a property called `name`.

1) Create a **constructor** to pass the name of the `oven`.
2) Create the methods `On()` and `Off()`. Display appropriate Messages when they are called.

Marks 2

ii) Create a **Receiver class** called `GarageGate`. The `GarageGate` class should have a **property** called `description`.

1) Create a **constructor** to pass the `description` of the Garage Gate.
2) Create the methods `Open()` and `Close()`. Display appropriate Messages when they are called.

Marks 2

iii) Create an **interface** called `Command` it should have a method called
`public void execute()`

Marks 1

iv) Create the **classes** `OvenOn` and `OvenOff` that implement the `Command` interface.

1) Both **classes** should have `Oven` type objects as properties.
2) The **constructors** of both classes should take `Oven` type objects as parameters.
3) Implement the `execute()` method, they should respectively call the `On()` and `Off()` methods of the `Oven` class object property.

Marks 5

v) Create the **classes** `GarageGateOpen` and `GarageGateClose` that implement the `Command` interface.

1) Both **classes** should have `GarageGate` type objects as properties.
2) The **constructors** of both classes should take `GarageGate` type objects as parameters.
3) Implement the `execute()` **method**, they should respectively call the `Open()` and `Close()` methods of the `GarageGate` class object property.

Marks 5

vi) Implement the `MobileUI` **class** that allows the user to customize the various buttons in the Mobile Application used for the Office Automation.

1) Have an array of `Command[]` objects to represent different commands that can be issued by the Mobile Application.
2) In the **constructor** declare the `Command[]` array to contain 6 command Buttons.
3) Implement a **method** called
`public void setCommand(int index, Command cmdObj)`
It should store the `cmdObj` under the correct `index` in the `Command[]` array.
4) Implement a method called
`public void commmandPressed(int index)`
This should execute the `command` given by the object.

Marks 4

vii) Implement a **class** called `MainApp` which contains the `main()` function.

1) Create an object of the `MobileUI` class.
2) Create an `Oven` object called `mainOven`.
3) Create a `GarageGate` object called `garageGate`
4) Create 4 command buttons to control turning the Oven `On` and `Off` and Opening and Closing the `GarageGate`.
5) Simulate the use of the Oven and the `GarageGate`.

Marks 4

Save the Project as **Ques04b**.

End of the Question Paper

IT2030 – OOP – PAPER – 1a

Student No : **Machine No :**

QUESTION	MARKS
1	
2	
3	
4	
TOTAL	