

**PROGRAMME :**

**DIPLOMA IN INFORMATION TECHNOLOGY**

**(DIGITAL TECHNOLOGY)(DDT)**

**COURSE:**

**DFP30243 - OBJECT ORIENTED PROGRAMMING**

|  |  |  |
| --- | --- | --- |
| **ASSESMENT** | **PROBLEM BASED TASK (PBT)** | |
| **NO** | **REGISTRATION NO** | **NAME** |
| 1. |  |  |
| 2. |  |  |
| 3. |  |  |
| **PROGRAMME** | DDT3 (YOUR CLASS NAME) | |

INSTRUCTIONS :

1. Answer **ALL** the questions.
2. Submission Date : ……………………………………………………

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **CODE / COURSE** | **DFP30243 –**  **OBJECT ORIENTED PROGRAMMING** | **PROBLEM BASED TASK** | **1** | |
| **PROGRAM / CLASS** |  | **DURATION** | **10 HOURS** | |
| **STUDENT’S NAME** |  | **CLO** | **1P** |  |
| **REG. NO.** |  | **TOTAL MARKS** | **/36** | |
| **LECTURER’S NAME** | **PN. HAZLEENA BINTI OSMAN /**  **PN. RODZIAH BINTI IBRAHIM /**  **PN SHARIZAN BINTI ABDUL JAMIL /**  **CIK NURAINI BINTI SHAMSAIMON** |

**CLO1P:** Construct Object Oriented Programming concept and exception handling in Java programming.

**Learning outcomes:**

Upon completion of this problem-based task, students should be able to:

1. Describe general UML elements.
2. Construct input and output statement in Java programs
3. Build java programs using variables, operators and Input/ Output Streams.
4. Build program using branching statements
5. Build an object in Java programs
6. Manipulate inheritance using ‘extends’ keyword.
7. Identify base and derived class.
8. Build the protected access specifier.
9. Construct overriding method
10. Construct abstract classes.
11. Build Java programs using exception handling.

**INSTRUCTION: Answer the question below.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| In a group of 2-3 students, develop a Java system to **calculate payment for Paradise Resort And Training Centre Booking System** based on **Figure 1: Class Diagram** and **Figure 2: Package Rate Details.**  **System Requirements:**   1. Log in into the system-using password: P@Qwerty$123. (Show payment receipt header after successful authentication or the application will exit if user key in the wrong password) 2. Input customer name and IC using BufferedReader. 3. Input **code of package type** (if user enter wrong code, prompt user to key-in the code again) 4. Input number of teambuilding participants (if user enter an invalid input, the system will handle the error by using exception handling) 5. Calculate totalPackagePrice by multiply package price and number of teambuilding participants 6. Calculate **30% discount** if number of teambuilding participants is **more than 80 person**. 7. Calculate totalPackagePrice after discount by substracting totalPackagePrice and discount. 8. Input payment from customer (if user enter an invalid input, the system will handle the error by using exception handling) 9. Calculate balance for customer. 10. Construct exception handling to handle the exception.  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | | PackageBooking | | +display(): void |  |  | | --- | | Booking<<abstract>> | | #calculateBooking(): void <<abstract>> |  |  | | --- | | Customer | | +name: String  +password: String  +ic: String  +code: String  + totalPackagePrice: double  +noOfParticipant: int  +custPayment: double  +packagePrice: double  +discount: double | | Customer (String name, String ic, double packagePrice, int noOfParticipant)  #calculateBooking():void  +display():void | |     **Figure 1**   |  |  |  |  | | --- | --- | --- | --- | | Code. | Teambuilding Package | Price (RM) /Per Pax | Discount 30%  Booking more than 80 person | | 001 | Government | 150.00 | | 002 | Corporate | 230.00 | | 003 | VIP | 330.00 |   **Figure 2**  **Hint:**   1. Apply an **overriding** concept in the program. Display () method on super class will be printed the payment receipt header. Display () method on child class will be printed receipt details. 2. Locate the main method in the Customer class to access the properties in their parent class using access specifier protected.   **Sample Output:**   1. **If user enter correct password**      1. **If user enter incorrect password**      1. **If user enter wrong input format for “No. Of Pax”,**   **system will handle the error by using exception handling**     1. **If user enter wrong input format for payment the**   **system will handle the error by using exception handling**     1. **If user enter all input correctly**     **Submission:**  1. Your PBT Report **softcopy (Word Document)** must consists:   * 1. Front page   2. Source code   3. Receipt Snapshot for **each Type of Room**   4. YouTube link/Video (PBT Presentation)  1. Upload **softcopy PBT Report and source code(.java files)** to CIDOS     **(36 Marks-Report, 20 Marks-PBT Presentation)** |

|  |  |  |  |
| --- | --- | --- | --- |
| **CLO** | **Criteria (PBT)** | **Marks** | **Total Marks** |
| **CLO1P** | **(A) Analyzing Information (Class Diagram)** | /4 | **/36** |
| **(B) Problem Solving** | /4 |
| **(C) Build program using branching statements**  **(Login & Package Choice)** | /4 |
| **(D) Polymorphism Concept (Overriding)** | /4 |
| **(E) Inheritance Concept (Multilevel Inheritance)** | /4 |
| **(F) Abstract Class and Method** | /4 |
| **(G) Exception Handling** | /4 |
| **(H) Efficiency and Output** | /4 |
| **(I) Documentation** | /4 |
| **CLO** | **Criteria (PBT Presentation)** | **Marks** | **Total Marks** |
| **CLO3A** | **Visual Appeal (Presentation Slide)**  **(Presentation Slide)**  **Comprehension**  **Presentation**  **Skills**  **Content**  **Preparedness/**  **Participation/**  **Group Dynamics** | /4 | **/20** |
| **Comprehension** | /4 |
| **Presentation Skills** | /4 |
| **Content** | /4 |
| **Preparedness/Participation/Group Dynamics** | /4 |

**Evaluation:**