**COMSATS** **University Islamabad, Lahore Campus**

**Department of Computer Science**

**Mid-Term – Semester Fall 2022**

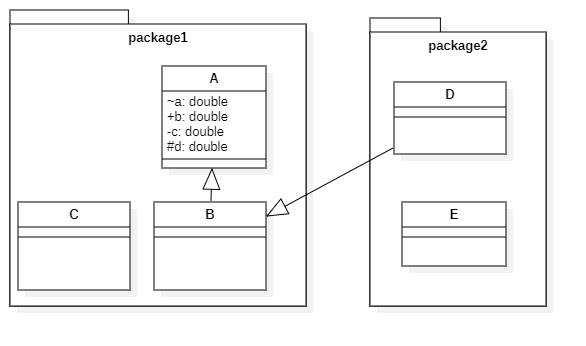
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course Title: | Object Oriented Programming | | | | Course Code: | | CSC241 | Credit Hours: | | 4(3,1) |
| Course Instructor/s: | Dr. M. Aksam Iftikhar | | | | Program Name: | | BCS | | | |
| Semester: | 3rd | Batch: | FA21-BCS | Section: | B, C | | Date: | | 16-11-2022 | |
| **Time Allowed:** | **90 Minutes** | | | | **Maximum Marks:** | | | | **30** | |
| Student’s Name: |  | | | | Reg. No. |  | | | | |
| **Important Instruction:**   * There are 4 questions in total. Solve first 3 questions on question paper, while the last question should be solved on answer sheet. | | | | | | | | | | |

**Q1. Recognize the most appropriate option for the following MCQs: [7 Marks]**

**(CLO1->C1, C2 (Knowledge, Understanding))**

1. Which of the following is the correct order of statements in a Java program?
   1. Class declaration, Package statement, Import Statement
   2. Import statement, Class declaration, Package statement
   3. Package statement, Import Statement, Class declaration
   4. Order of all above mentioned elements does not matter
2. Select the correct option to create an object of a class named Child inherited by class Parent
   1. Child c1 = new Child;
   2. Child c1 = new Child();
   3. Parent c1 = new Child();
   4. Both b and c
3. Which of these keywords can be used to prevent Method overriding?
   1. static
   2. constant
   3. protected
   4. final
4. Which of these is correct way of calling a no-parameter constructor of superclass A inside subclass B constructor?
   1. super(void);
   2. super.();
   3. super.A();
   4. super();
5. Super reference MUST be used in constructor of the sub class as a very first statement of the constructor, otherwise a complie-time error occurs.
6. True
7. False
8. Static members and methods can be accessed using the class name.
   1. True
   2. False
9. The use of @override keyword is must while overriding a method, otherwise, a compile time error occurs.
10. True
11. False

**Q 2: Consider the following UML class diagram and answer the questions given below. (CLO1->C2 (Understanding)) [5 Marks]**



1. Consider the attributes of class A and answer by filling with Y and N where fields are accessible or not.

|  |  |  |  |
| --- | --- | --- | --- |
|  | b | c | d |
| Class C can access attributes of A |  |  |  |
| Class D can access attributes of A |  |  |  |
| Class E can access attributes of A |  |  |  |

1. Can class C access field ‘d’ of class A? Yes / No

**Q 3: Show the output of following programs and if there is, any syntax/logical error in the code mention that. [10 Marks]**

**(CLO1->C3 (Analyzing))**

|  |  |  |
| --- | --- | --- |
| **No** | **Code** | **Output/Error** |
|  | **package** MidTermExam; **public class** MidTermExam {  **public int id**;  **public** String **name**;  **public boolean status**;  **public** MidTermExam(**int** id) {  **this**.**id** = id;  System.***out***.println(**"Hello"**);  }  **public** MidTermExam(**int** id, String name) {  **this**(3);  **this**.**id** = id;  **this**.**name** = name;  System.***out***.println(**"Mid Term Exam"**);  }   **public** MidTermExam(**int** id, String name,**boolean** status) {  **this**(2,**"Ayesha"**);  System.***out***.println(**"Best of luck"**);  }   **public static void** main(String[] args) {  MidTermExam m1 = **new** MidTermExam(4,**"Saad"**,**true**);  System.***out***.println(m1.**id**);  System.***out***.println(m1.**name**);  System.***out***.println(m1.**status**);  } } |  |
|  | **class** Super {  **int id**;  **public** Super() {  System.***out***.println(**"No arg Cons in parent class"**);  }   **public** Super(**int** id) {  **this**.**id** = id;  System.***out***.println(**"One-arg Cons in parent class"**);  } } **class** Sub **extends** Super {  **public** String **name**;   **public** Sub(String name) {  **this**.**name** = name;  System.***out***.println(**"One arg constructor in child class"**);  } } **public class** MidTermExam {  **public static void** main(String[] args) {  Super s1 = **new** Sub(**"Zara"**);  } } |  |
|  | **public class** MidTermTest {  **public static void main**(String[] args) {  **int** array[][] = **new** **int**[3][];  array[0] = **new int**[]{0,2,4,6};  array[1] = **new int**[]{1,3,5,7,9};  array[2] = **new int**[]{10,20,30};   **for** (int i = 0; i < array.**length**-1; i+=1)  {  **for** (int j = 0; j < array[i].**length**-1;j+=2) {  System.*out*.printf("%d ",array[i][j]);  }  System.*out*.println();  }  } } |  |

**Q 4:** Answer the questions given below considering Date, Person and Employee classes below. **[8 Marks]**

(CLO1->C5, C6 (Creating))

|  |
| --- |
| **class** Date {  **public** **int day**;  **public** **int month**;  **public** **int year**;  **public** Date(**int** day, **int** month, **int** year) {  **this**.**day** = day;  **this**.**month** = month;  **this**.**year** = year;  } } |
| **public abstract class** Person {  **public** String **name**;  **public** Date **dob**;  **public** Person(String name, Date dob) {  **this**.**name** = name;  **this**.**dob** = dob;  } } |
| **public class** Employee **extends** Person{  **public** Employee(String name, Date dob) {  **super**(name, dob);  } } |
| **import** java.util.ArrayList; **public class** Demo {  **public static void** main(String[] args) { // your code part a }  **Public static** Employee copyEmployee(Employee e1) { // your code part b }  } |

1. Inside the main method, write down the code to create an ArrayList to store 1 person and 2 Employee objects. All these objects will have date of birth 01 Jan, 2001.
2. Complete the method copyEmployee, so that it creates and returns a *deep copy* of e1.
3. Rewrite the Employee class so that it now implements the Comparable interface. The compareTo method inherited from the Comparable should compare two Employee objects based on only the year of their date of birth.