**Assignment # 03**

**Question 01:**  **(10 marks)**

1. Whenever the derived class inherits the base class`s data members there is a possibility that base class data members are similar to derived class data members and JVM gets an ambiguity, how we can resolve this ambiguity?

For Example:

class A {

int x;

}

class B extends A{

int x;

}

1. What is “Abstract Class” in java? Write down the rules required for it.
2. Whether it is mandatory to have an abstract method in abstract class? If not why such design is required?
3. What is the difference between Abstract Class and Interface?
4. Why we used interface? Write down the properties of interface.

**Question 02:**  **(10 marks)**

1. Create a class **Heater** that contain a single integer field called **temperature**. Define a **constructor** that takes no parameters. The temperature field should be set to the **value 15 in the constructor**. Write setter functions called warmer and cooler, whose effect is to increase or decrease the value of the temperature by 5 respectively. Write down a getter method to return the value of temperature.
2. Let's assume we want to create an app with an interrogator that can convince any animal to talk. We will create an Interrogator class that is responsible for convincing the animals to talk. We don't want to write a method for each type of animal: For Example **convinceDogToTalk**(Dog dog) for Dog, **convinceCatToTalk**(Cat cat) for Cat, and so on. We would prefer one general method that would accept any animal. How can we do this? Write down the required code for it.

**Question 03:**  **(15 marks)**

We want to develop a graphical editor (like Windows Paint application) that should be able to draw different geometric shapes such as line, circle, and triangle. User should be able to select shape of his/her choice and by dragging the mouse onto the screen user should be able to draw the selected shape onto the canvas (Painting Area on Screen). User should also be able to select, move, resize, or rotate a shape. To do so, editor should provide the user with a menu listing different commands. Individual shapes should also be grouped together and behave as single shape.

Considering the case study provided above, perform the following tasks:

1. Identify objects/classes using domain knowledge.
2. Extract properties and Operations of objects using domain knowledge.
3. Identify relationships between objects.
4. Draw a UML class diagram according to your recommended object-oriented solution.
5. Implement the code skeleton of class diagram created in question (d), using java programming language.

**Question 04:**  **(5 marks)**

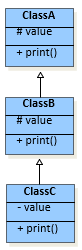
Identify, which OOP concept is represented in the given scenarios?

**Options:** Abstraction, Encapsulation, Inheritance & Polymorphism

|  |  |
| --- | --- |
| **Scenario** | **OOP Concept** |
| 1. **The wrapping up of data and functions into a single unit is called** |  |
| 1. **In object-oriented programming, new classes can be defined by extending existing classes.** |  |
| 1. Every time you log into your email account( GMail, Yahoo, Hotmail or official mail), you have a whole lot of processes taking place in the backend, that you have no control over. So your password, would probably be retrieved in an encyrpted form, verified and only then you are given access. You do not have any control, over how the password is verified, and this keeps it safe from misuse. |  |
| 1. When you log into your email, compose and send a mail. Again there is a whole lot of background processing involved, verifying the recipient, sending request to the email server, sending your email. Here you are only interested in composing and clicking on the send button. What really happens when you click on the send button, is hidden from you. |  |
| 1. Automatic cola vending machine is a class. It contains both data i.e. Cola can and operations i.e. service mechanism and they are integrated under a single unit Cola Vending Machine |  |

**Question 05:**  **(10 marks)**

|  |  |
| --- | --- |
| 1. What will be the output of the below program? | 1. Ali has written the code provided below, in Java programming language. The code is showing compile time error. Can you identify what mistake he has done? |
| class A  {      {          System.out.println(1);      }  }    class B extends A  {      {          System.out.println(2);      }  }    class C extends B  {      {          System.out.println(3);      }  }    public class MainClass  {      public static void main(String[] args)      {          C c = new C();      }  } | class X  {  int x;      //Other Class X Members  }    class Y  {  int y;      //Other Class Y Members  }    class Z extends X, Y  {  int z;      //Other Class Z Members  } |

**Question 06:**  **(6 marks)**

**class ClassA {**

**protected int value;**

**public ClassA() { }**

**public ClassA(int val) { value = val; }**

**public void print() {**

**System.out.println("Class A: value = " + value);**

**}**

**}**

ClassA.java

ClassB.java

**class ClassB extends ClassA {**

**protected int value;**

**public ClassB() { }**

**public ClassB(int val) {**

**super.value = val – 1;**

**value = val;**

**}**

**public void print() {**

**super.print();**

**System.out.println("Class B: value = " + value);**

**}**

**}**

ClassC.java

**final class ClassC extends ClassB {**

**private int value;**

**public ClassC() { }**

**public ClassC(int val) {**

**super.value = val – 1;**

**value = val;**

**}**

**public void print() {**

**super.print();**

**System.out.println("Class C: value = " + value);**

**}**

**}**

**public class TestSubclasses {**

**public static void main(String[] args) {**

**ClassA objA = new ClassA(123);**

**ClassB objB = new ClassB(456);**

**ClassC objC = new ClassC(789);**

**objA.print(); System.out.println("---------");**

**objB.print(); System.out.println("---------");**

**objC.print();**

**}**

**}**

TestSubclasses.java

**Write the output of the code provided above.**

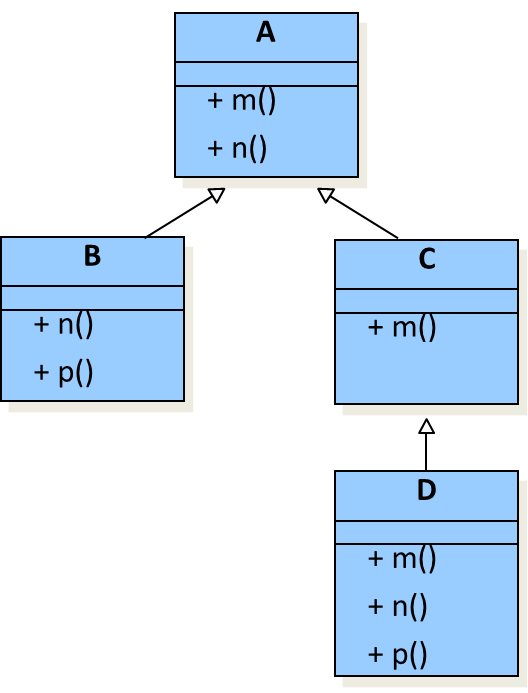
**Question 06:**  **(07 marks)**

Cosider the UML Class diagram provided below, Assume all methods print out message of the form <class name>,<method name>, infront of the code fragments in the table provided below.

e.g.: method m() in class A prints out “A.m”.

If a class overrides an inherited method, the method’s name will appear in the class icon. Otherwise, the inherited method remains unchanged in the subclass.

For each code fragment below, indicate whether:

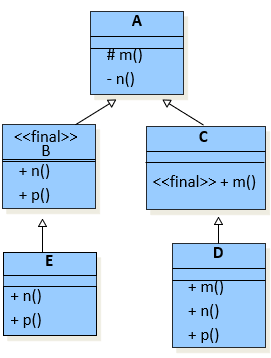
* ****The code will cause compilation error, and briefly explain; or
* The code can compile and run. Supply the execution result.

**Examples:**

|  |  |  |
| --- | --- | --- |
| **Code fragment (example)** | **Compilation error? Why?** | **Execution result** |
| **A a = new A();**  **a.m();** |  | **A.m** |
| **A a = new A();**  **a.k();** | **Method k() not defined in class A** |  |

|  |  |  |
| --- | --- | --- |
| **Code fragment** | **Compilation error?** | **Execution result** |
| **A a = new C();**  **a.m();** |  |  |
| **B b = new A();**  **b.n();** |  |  |
| **A a = new B();**  **a.m();** |  |  |
| **A a;**  **C c = new D();**  **a = c;**  **a.n();** |  |  |
| **B b = new D();**  **b.p();** |  |  |
| **C c = new C();**  **c.n();** |  |  |
| **A a = new D();**  **a.p();** |  |  |

**Question 07:**  **(7 marks)**

****

Cosider the following UML Class diagram, Assume all methods print out message of the form <class name>,<method name>, infront of the code fragments in the table provided below.

e.g.: method m() in class A prints out “A.m”.

If a class overrides an inherited method, the method’s name will appear in the class icon. Otherwise, the inherited method remains unchanged in the subclass.

For each code fragment below, indicate whether:

* The code will cause compilation error, and briefly explain; or
* The code can compile and run. Supply the execution result.

|  |  |  |
| --- | --- | --- |
| **Code fragment** | **Compilation error?** | **Execution result** |
| **A a = new C();**  **a.m();** |  |  |
| **B b = new A();**  **b.n();** |  |  |
| **E e = new E();**  **e.m();** |  |  |
| **D d = new D();**  **d.M();** |  |  |
| **B b = new D();**  **b.p();** |  |  |
| **E e = new E();**  **e.m();** |  |  |
| **A a = new D();**  **a.p();** |  |  |