**Homework 1**

For problems 1 through 4, explain why the code as shown is almost certainly not what the programmer intended, and how it should be fixed to work the way the programmer probably had in mind.

1. (10 pts) What is wrong with the following program and how should it be fixed?

1  public class MyClassA {  
2    int v = 12;  
3   
4    public MyClassA (int pV) {  
5      v = pV;  
6    }   
7   
8    public static void main (String args []) {  
9      MyClassA m = new MyClassA ();  
10   } // end main  
11 } // end class MyClassA

**SOLUTION:**

**The problem with this program is that there is no default constructor. Instead the current constructor on line 4 requires an int value which is not included within the class object on line 9. There are two ways to fix this:**

**Number 1 – Include an int variable in the class object instance on line 9**

**ex: MyClassA m = new MyClassA(12);**

**Number 2 – Turn the current constructor into a default constructor by removing the int value requirement on line 4.**

**ex: public MyClassA() {}**

2. (10 pts) What is wrong with the following program and how should it be fixed?

1  public class MyClassB {  
2    int v = 12;  
3   
4    public void MyClassB (int pV) {  
5      v = pV;  
6    }   
7   
8    public static void main (String args []) {  
9      MyClassB m = new MyClassB (23);  
10   } // end main  
11 } // end class MyClassB

**SOLUTION:**

**The problem with this program is that the keyword “void” is making the function on line 4 behave as a method as opposed to a constructor. The easiest way to fix this is to just remove the “void” keyword”.**

**ex: public MyClassB(int pV){}**

3. (10 pts) What is wrong with the following program and how should it be fixed?

1   public class MyClassD {  
2     public static void main (String args []) {  
3       MyClassC m = new MyClassC (23);  
4     } // end main  
5   } // end class MyClassD  
6   
7   class MyClassC {  
8     int v = 12;  
9   
10    public MyClassC (int pV) {  
11      int v = pV;  
12    }   
13   
14  } // end class MyClassC

**SOLUTION:**

**The problem with this program is that the variable v has already been initialized on line 8 and doesn’t need to be initialized again on line 11. To fix this, remove the int keyword before the v on line 11.**

**ex: v =pV;**

4. (10 pts) What is wrong with the following program and how should it be fixed?

1   public class MyClassE {  
2     public static void main (String args []) {  
3       MyClassF m = new MyClassF (23);  
4     } // end main  
5   } // end class MyClassE  
6   
7   class MyClassF {  
8     int v = 12;  
9   
10    private MyClassF (int pV) {  
11      v = pV;  
12    }   
13   
14  } // end class MyClassF

**SOLUTION:**

**The problem with this program is that the class MyClassE cannot call the constructor in class MyClassF because the constructor has the “private” keyword before it, limiting access solely to the class MyClassF. To fix this, all one has to do it change the constructor access from “private” to “public” on line 10, to allow other classes access.**

**ex: public MyClassF (int pV) {}**

5. (10 pts) Given all the problems identified in problems 1 through 4, explain in detail why the following code works, ie, compiles without errors or warnings.

1  public class MyClassG {  
2    public static void main (String args []) {  
3      MyClassH m = new MyClassH (23, true);  
4    } // end main  
5  } // end class MyClassG  
6   
7  class MyClassH {  
8    int v = 12;  
9   
10   public MyClassH (int x, boolean b) {  
11     this (x);  
12   }   
13   
14   private MyClassH (int pV) {  
15     v = pV;  
16   }   
17   
18 } // end class MyClassH

**SOLUTION:**

**This program compiles without errors because the constructors are well initialized and the variables are declared and used in the right places.**

6. (10 pts) Explain why the following class hierarchy is not reasonable:

* DefenseDepartment
  + General
    - Private

**SOLUTION:**

**The current class hierarchy is not reasonable because a Private is not a type of General. As such it would not be an appropriate subclass and would instead fall under the class DefenseDepartment as both a General and a Private work for the DefenseDepartment. Should be organized like this:**

* **DefenseDepartment**
  + **General**
  + **Private**

7. (10 pts) Give at least one example of a reasonable field for each of the following classes in the following class hierarchy. Be sure that the field is at the right level in the hierarchy.

* Vehicle
  + Car
  + Airplane
    - Passenger
    - Fighter
    - Bomber
  + SpaceShip

**SOLUTION:**

**Field for Vehicle Class:**

**private String vehicleType;**

**Field for Car Class:**

**private String carType;**

**Field for Airplane Class:**

**private String airplaneType;**

**Field for Passenger Class:**

**private int numSeats;**

**Field for Fighter Class:**

**private double maxVelocity;**

**Field for Bomber Class:**

**private int numBombs;**

**Field for SpaceShip Class:**

**private Boolean artificialGravityOn;**

8. (10 pts) Give at least one example of a reasonable method for each of the following classes in the following class hierarchy. Be sure that the method  is at the right level in the hierarchy. Constructors, getters and setters don't count for this problem.

* Vehicle
  + Car
  + Airplane
    - Passenger
    - Fighter
    - Bomber
  + SpaceShip

**SOLUTION:**

**Method for Vehicle Class:**

**public static Vehicle(String vehicleType, int numPassengers)**

**Method for Car Class:**

**public static Car(String make, String model, String color, int year)**

**Method for Airplane Class:**

**Public static Airplane(String airplaneType, double maxVelocity)**

**Method for SpaceShip Class:**

**Public static SpaceShip(int passengerCapacity, Boolean artificialGravityOn)**

9. (10 pts) Are a Private and a Platoon in an encapsulation or an inheritance relationship? Explain

**SOLUTION:**

**A Private and a Platoon are in an encapsulation type of relationship because they are both smaller units that make a larger unit (ex: Battalion). As such, they can both be encapsulated into the same superclass.**   
  
10. (10 pts) Present reasonable parent and child classes for the class Tree (the biological kind). Give a short explanation for why the classes you are proposing are in good parent-child relationships.

**SOLUTION:**

* **Flora**
  + **Tree**
    - **CedarTree**

**The parent class of the class Tree is Flora, which is a term that describes all plant-life in an environment, in which trees are included. The child class to the class Tree is CedarTree which is a type of tree.**

**Grading Rubric:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Meets** | **Does not meet** |
| Problem 1 | **10 points** Explains why the code as shown is almost certainly not what the programmer intended.  Explains how it should be fixed to work the way the programmer probably had in mind. | **0 points** Does not explain why the code as shown is almost certainly not what the programmer intended.  Does not explain how it should be fixed to work the way the programmer probably had in mind. |
| Problem 2 | **10 points** Explains why the code as shown is almost certainly not what the programmer intended.  Explains how it should be fixed to work the way the programmer probably had in mind. | **0 points** Does not explain why the code as shown is almost certainly not what the programmer intended.  Does not explain how it should be fixed to work the way the programmer probably had in mind. |
| Problem 3 | **10 points** Explains why the code as shown is almost certainly not what the programmer intended.  Explains how it should be fixed to work the way the programmer probably had in mind. | **0 points** Does not explain why the code as shown is almost certainly not what the programmer intended.  Does not explain how it should be fixed to work the way the programmer probably had in mind. |
| Problem 4 | **10 points** Explains why the code as shown is almost certainly not what the programmer intended.  Explains how it should be fixed to work the way the programmer probably had in mind. | **0 points** Does not explain why the code as shown is almost certainly not what the programmer intended.  Does not explain how it should be fixed to work the way the programmer probably had in mind. |
| Problem 5 | **10 points** Given all the problems identified in problems 1 through 4, explains in detail why the code works, ie, compiles without errors or warnings. | **0 points** Given all the problems identified in problems 1 through 4, does not explain in detail why the code works, ie, compiles without errors or warnings. |
| Problem 6 | **10 points** Explains why the class hierarchy is not reasonable. | **0 points** Does not explain why the class hierarchy is not reasonable. |
| Problem 7 | **10 points** Gives at least one example of a reasonable field for each of the classes.  The field is at the right level in the hierarchy. | **0 points** Does not give at least one example of a reasonable field for each of the classes.  The field is not at the right level in the hierarchy. |
| Problem 8 | **10 points** Gives at least one example of a reasonable method for each of the classes.  The method is at the right level in the hierarchy.  Does not include constructors, getters and setters. | **0 points** Does not give at least one example of a reasonable method for each of the classes.  The method is not at the right level in the hierarchy.  Includes constructors, getters and setters. |
| Problem 9 | **10 points** Explains inheritance and encapsulation correctly and in sufficient detail given the example provided. | **0 points** Does not explain inheritance and encapsulation correctly and in sufficient detail given the example provided. |
| Problem 10 | **10 points** Presents reasonable parent and child classes for the class Tree.  Gives a short explanation for why the classes you are proposing are in good parent-child relationships. | **0 points** Does not present reasonable parent and child classes for the class Tree.  Does not give a short explanation for why the classes you are proposing are in good parent-child relationships. |