**Homework 2**

1. (10 pts) What are the diagrams defined in the UML Standard. Give a one or two sentence description of each one.

**UML Diagrams are divided into two broad categories: Structural Diagrams and Behavioral Diagrams.**

**Structural Diagrams – Represent the static aspect of the system, which forms the main structure and are therefore stable.**

**Class Diagram – Most common UML diagrams. They represent the object-oriented view of a system which consists of classes, interfaces associations, and collaboration.**

**Object Diagram – Set of objects and their relationships that describe an instance of a class diagram.**

**Component Diagram – Represents the relationships between components such as classes, interfaces, and collaborations to better see the implementation of a system.**

**Deployment Diagram – Used for visualizing the physical presence of nodes and their relationships.**

**Structural Diagrams – Represent the dynamic aspect of the system by describing the changing/moving parts of a system.**

**Use Case Diagram – Represent the functionality of a system by describing the relationships among functionalities and their internal/external controllers (actors).**

**Sequence Diagram – Used to visualize the sequence of calls from object to another in a system to perform a specific functionality.**

**Collaboration Diagram – Represents the structural organization of a system and the messages that are sent and received between objects via links.**

**Statechart Diagram – Describes the state of change of a class, interface, and collaborations in a system as a result of internal and external factors.**

**Activity Diagram – Describes the flow of controls between functions in a system. The activity can be either sequential, concurrent, or branched.**

2. (10 pts) Given the following code, how should the toString methods in the classes H2ClassA and H2ClassB be written to give the indicated output and take advantage of the natural toString method in H2ClassB?

1  import java.util.ArrayList;  
2   
3  public class H2ClassA {  
4    ArrayList <H2ClassB> list = new ArrayList <H2ClassB> ();

**EDIT:**  
**5 // You want to add the toString Method within class H2ClassA**

**public String toString() {**

**// initialize output string**

**String output = "";**

**//Using for-loop add the list with the numbers**

**for (int i = 0; i < list.size(); ++i) {**

**// add numbers to output string**

**output = output.concat(Integer.toString(list.get(i).x))+" ";**

**}**

**return output;**

**}**  
6    public static void main (String args []) {  
7      H2ClassA y = new H2ClassA ();  
8      int [] v = {4, 3, 7, 5, 99, 3};  
9      for (int m: v)   
10       y.list.add (new H2ClassB (m));  
11     System.out.println (y);  
12   } // end main  
13   
14 } // end class H2ClassA  
15   
16 class H2ClassB {  
17   int x;  
18   H2ClassB (int a) { x = a;}  
19 } // end H2ClassB

OUTPUT:

4 3 7 5 99 3

3. (10 pts) How can the following code be corrected? Give at least two good answers.

1 public class H2ClassC {  
2   H2ClassC (int a) {}  
3 } // end class H2ClassC  
4   
5 class H2ClassD extends H2ClassC{  
6 } // end class H2ClassD

**1st Answer: Remove the constructor is class H2ClassC and simple declare the variable “a” as an integer instead. Then create a constructor in class H2ClassD and use the super keyword to call the parameter from the superclass H2ClassC.**

**EDIT:**

**public class H2ClassC {**

**// remove constructor, and simple declare variable**

**int a;**

**} // end class H2ClassC**

**class H2ClassD extends H2ClassC {**

**// create constructor for H2ClassD w/ super keyword to call parameter**

**H2ClassD(int a) {**

**super();**

**}**

**} // end class H2ClassD**

**2nd Answer: Create a constructor for class H2ClassD and call the parameter using super keyword**

**EDIT:**

**public class H2ClassC {**

**H2ClassC (int a) {}**

**} // end class H2ClassC**

**class H2ClassD extends H2ClassC {**

**// create a constructor and call the parameter w/ super keyword**

**H2ClassD(int a) {**

**super(a);**

**}**

**} // end class H2ClassD**

4. (10 pts) Why does the following code give a compiler error? How should it be fixed?

1  public class H2ClassE {  
2    int x, y, z;  
3   
4    H2ClassE (int a) {  
5      x = a;  
6      this (5, 12);  
7    }  
8   
9    H2ClassE (int b, int c) {  
10     y = b;  
11     z = c;  
12   }  
13 } // end class H2ClassE

**Answer: The code gives a compiler error because the constructor call “this (5,12)” needs to be the first statement in the constructor. To fix the code you simply have to swap it with “x=a”.**

**EDIT:**

**public class H2ClassE {**

**int x, y, z;**

**H2ClassE (int a) {**

**this (5, 12);**

**x = a;**

**}**

**H2ClassE (int b, int c) {**

**y = b;**

**z = c;**

**}**

**} // end class H2ClassE**

5. (10 pts) What is wrong with the following declaration? How should it be fixed?

public static final int myNumber = 17.36;

**Answer: The problem with the declaration is that the declared variable type (int) doesn’t match the variable type (double) of the variable (17.36). To fix this, the variable declaration should be:**

**EDIT:**

**public static final double myNumber = 17.36;**

6. (10 pts) What is wrong with the following code? How should it be fixed?

1 public class H2ClassG {  
2   final int x;  
3   
4   H2ClassG () {}  
5   H2ClassG (int a) {x = a;}  
6 } // end class H2ClassG

**Answer: The problem with the code is that the variable “x” is declared with the access keyword “final”, which means that that value of “x” cannot change. However, it is later being assigned the value of “a”. To fix this, simply remove the “final” access keyword.**

**EDIT:**

**public class H2ClassG {**

**int x=0;**

**H2ClassG () {}**

**H2ClassG (int a) {x = a;}**

**} // end class H2ClassG**

7. (10 pts) What is wrong with the following code? How should it be fixed?

1 public class H2ClassH {  
2   final int x;  
3   
4   int H2ClassH () {  
5     if (x == 7) return 1;  
6     return 2;  
7   } // end  
8 } // end class H2ClassH

**Answer: The problem with the code is that a variable with the final access keyword has to be initialized upon declaration, which in this example it is not. To fix this, the variable “x” needs to be initialized with a value, such as the integer 7.**

**EDIT:**

**public class H2ClassH {**

**final int x = 7;**

**int H2ClassH () {**

**if (x == 7) return 1;**

**return 2;**

**} // end**

**} // end class H2ClassH**

8. (10 pts) What is wrong with the following code? x should be given a value of 24. What are two ways this can be legally accomplished?

1 public class H2ClassI {  
2   final int x;  
3   
4   public static void main (String args []) {  
5     H2ClassI h = new H2ClassI ();  
6     h.x = 24;  
7   } // end main  
8 } // end class H2ClassI

**1st Answer: Remove the final access keyword so that the variable “x” can be initialized elsewhere than the variable declaration.**

**EDIT:**

**public class H2ClassI {**

**public int x;**

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

**H2ClassI h = new H2ClassI ();**

**h.x = 24;**

**} // end main**

**} // end class H2ClassI**

**2nd Answer: Create a constructor and initialize the variable “x” inside of it with the value of 24.**

**EDIT:**

**public class H2ClassI {**

**final int x;**

**H2ClassI() {**

**x = 24;**

**}**

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

**H2ClassI h = new H2ClassI ();**

**} // end main**

**} // end class H2ClassI**

9. (10 pts) What is wrong with the following code? Give two effective ways to fix it.

1  import javax.swing.\*;  
2  import java.awt.event.\*;  
3   
4  public class H2ClassJ extends JFrame {  
5    public static final long serialVersionUID = 22;  
6   
7    public H2ClassJ () {  
8      addMouseListener (new MouseListener () {  
9        public void mouseClicked (MouseEvent e) {}  
10     });  
11   } // end constructor  
12   
13 } // end class H2ClassJ

**1st Answer: The problem with the code is that the listener interface utilizes five abstract methods, so the class H2ClassJ either needs to be declared as an abstract class as seen here:**

**EDIT:**

**import javax.swing.\*;**

**import java.awt.event.\*;**

**public class H2ClassJ extends JFrame {**

**public static final long serialVersionUID = 22;**

**public H2ClassJ () {**

**addMouseListener (new MouseListener () {**

**public void mouseClicked (MouseEvent e) {}**

**public void mousePressed (MouseEvent e) {}**

**public void mouseReleased (MouseEvent e) {}**

**public void mouseEntered (MouseEvent e) {}**

**public void mouseExited (MouseEvent e) {}**

**});**

**} // end constructor**

**} // end class H2ClassJ**

**2nd Answer: Or implement all abstract methods of the MouseListener interface.**

**EDIT:**

**import javax.swing.\*;**

**import java.awt.event.\*;**

**public class H2ClassJ extends JFrame {**

**public static final long serialVersionUID = 22;**

**public H2ClassJ () {**

**addMouseListener (new MouseAdapter() {**

**public void mouseClicked (MouseEvent e) {}**

**});**

**} // end constructor**

**} // end class**

10. (10 pts) Why does the following code give a compiler warning? (Use javac -Xlint) How should it be fixed?

1 import javax.swing.\*;  
2   
3 public class H2ClassK {  
4   String [] sa = {"a", "b", "c"};  
5   JComboBox jcbA = new JComboBox (sa);  
6 } // end class H2ClassK

**Answer: The code gives a compiler warning because the declared variables should be declared as static variables. This is because variables that are declared outside of the main function cannot be accessed in the main function.**

**EDIT:**

**import javax.swing.\*;**

**public class H2ClassK {**

**// Static Variable Declaration**

**static String [] *sa* = {"a", "b", "c"};**

**static JComboBox *jcbA* = new JComboBox (*sa*);**

**}**

**Grading Rubric:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Meets** | **Does not meet** |
| Problem 1 | **10 points** Gives a one or two sentence description of each standard UML diagram. | **0 points** Does not give a one or two sentence description of each standard UML diagram. |
| Problem 2 | **10 points** Explains how the toString methods in the classes H2ClassA and H2ClassB be written to give the indicated output and take advantage of the natural toString method in H2ClassB. | **0 points** Does not explains how the toString methods in the classes H2ClassA and H2ClassB be written to give the indicated output and take advantage of the natural toString method in H2ClassB. |
| Problem 3 | **10 points** Provides at least two good answers explaining how the code can be corrected. | **0 points** Does not provide at least two good answers explaining how the code can be corrected. |
| Problem 4 | **10 points** Explains why the code gives a compiler error.  Explains how the code should be fixed. | **0 points** Does not explain why the code gives a compiler error.  Does not explain how the code should be fixed. |
| Problem 5 | **10 points** Explains what is wrong with the declaration.  Explains how the code should be fixed. | **0 points** Does not explain what is wrong with the declaration.  Does not explain how the code should be fixed. |
| Problem 6 | **10 points** Explains what is wrong with the code.  Explains how the code should be fixed. | **0 points** Does not explain what is wrong with the code.  Does not explain how the code should be fixed. |
| Problem 7 | **10 points** Explains what is wrong with the code.  Explains how the code should be fixed. | **0 points** Does not explain what is wrong with the code.  Does not explain how the code should be fixed. |
| Problem 8 | **10 points** Explains what is wrong with the code.  Explains two ways x could be given a values of 24 legally. | **0 points** Does not explain what is wrong with the code.  Does not explain two ways x could be given a values of 24 legally. |
| Problem 9 | **10 points** Explains what is wrong with the code.  Explains 2 effective ways the code could be fixed. | **0 points** Does not explain what is wrong with the code.  Does not explain 2 effective ways the code could be fixed. |
| Problem 10 | **10 points** Explains why the code gives a compiler warning.  Explains how it should be fixed. | **0 points** Does not explain why the code gives a compiler warning.  Does not explain how it should be fixed. |