

Date:20-April-2005
Time:9:00a.m. - 12:00p.m.
Permitted Aids:None

CS 132

Final Exam

Winter 2005

Signature: _____

Place sticker here.

Last Name: _____

First Name: _____

ID: _ _ _ _ _

Instructions: (Read carefully before the exam begins):

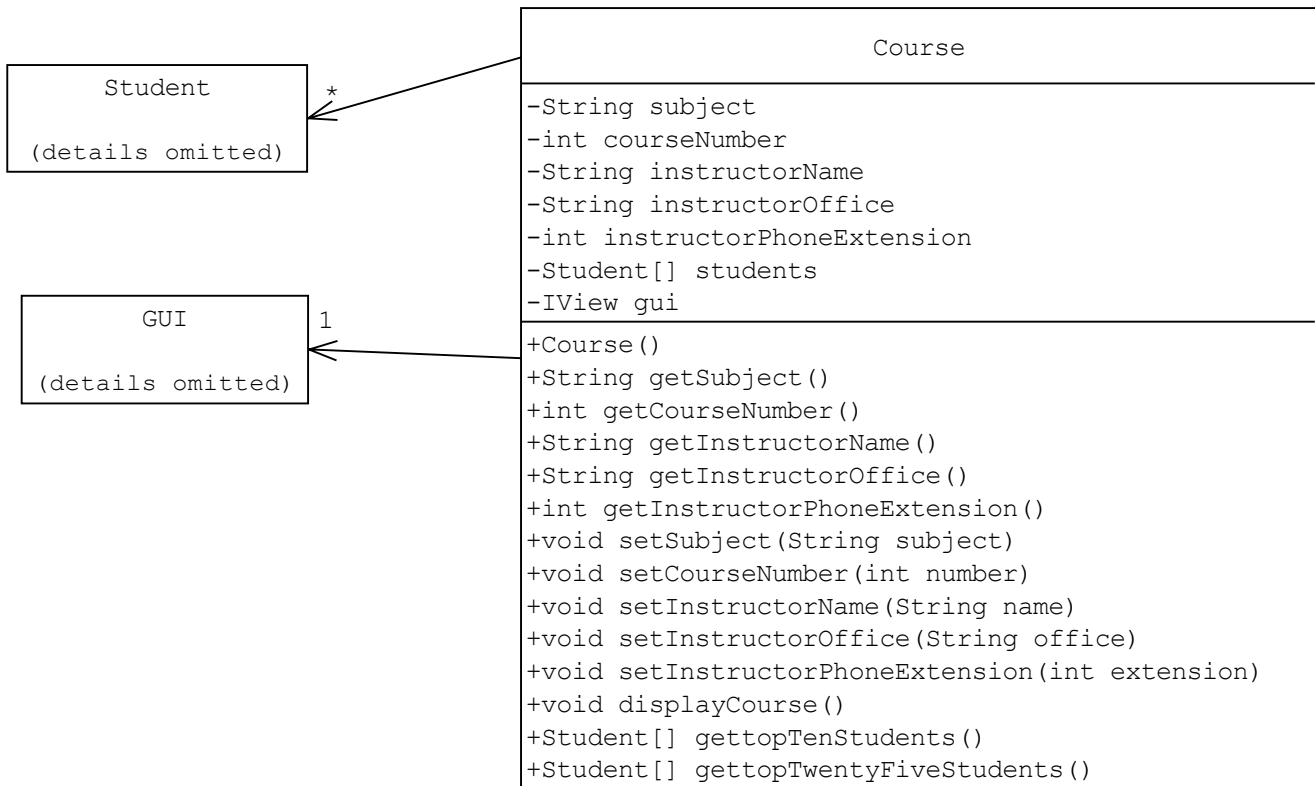
1. Before you begin, make certain that you have one Exam Booklet with pages numbered 1 – 20 printed double-sided and one Reference Booklet with pages numbered 1 – 3 printed double-sided.
2. The marks assigned to each question are shown at the beginning of the question; use this information to organize your time effectively.
3. Place all your answers in the spaces provided on these pages. Use the last page for extra space if you need it, but **clearly indicate this** if you do so.
4. You do not need to document your code unless it is specifically required by the question.
5. Questions will **not** be interpreted. Proctors will only confirm or deny errors in the questions. If you consider the wording of a question to be ambiguous, state your assumptions clearly and proceed to answer the question to the best of your ability. You may not trivialize the problem in your assumptions.
6. Cheating is an academic offense. Your signature on this exam indicates that you understand and agree to the University's policies regarding cheating on exams.

| Question | Max. Marks | Marks Given | Grader Initials |
|-----------------|------------|-------------|-----------------|
| Page 2, 3 | 8 | | |
| Page 5, 6 | 12 | | |
| Page 7 | 5 | | |
| Page 9 | 9 | | |
| Page 10 | 6 | | |
| Page 11, 12 | 8 | | |
| Page 13 | 7 | | |
| Page 14 | 12 | | |
| Page 15 | 6 | | |
| Page 16 | 7 | | |
| Page 17, 18, 19 | 10 | | |
| Page 20 | 8 | | |
| Total | 98 | | |

Object-Oriented Heuristics [8 marks]

A UML class diagram has been developed for the next “improvement” of Quest. A co-op student from another university did a poor job designing the `Course` class. It has many flaws and violates each of the following four object-oriented heuristics.

1. Use constructors to initialize an object’s state.
 2. Delegate work to helper class.
 3. Use parameters and overloading to combine several methods into a higher-level abstraction.
 4. Separate the view from the model.
- a. [4] Clearly label where each heuristic (1, 2, 3, and 4) is violated in the `Course` UML class diagram. There will be some flexibility in the marking of this question but your answer must be **clear**.

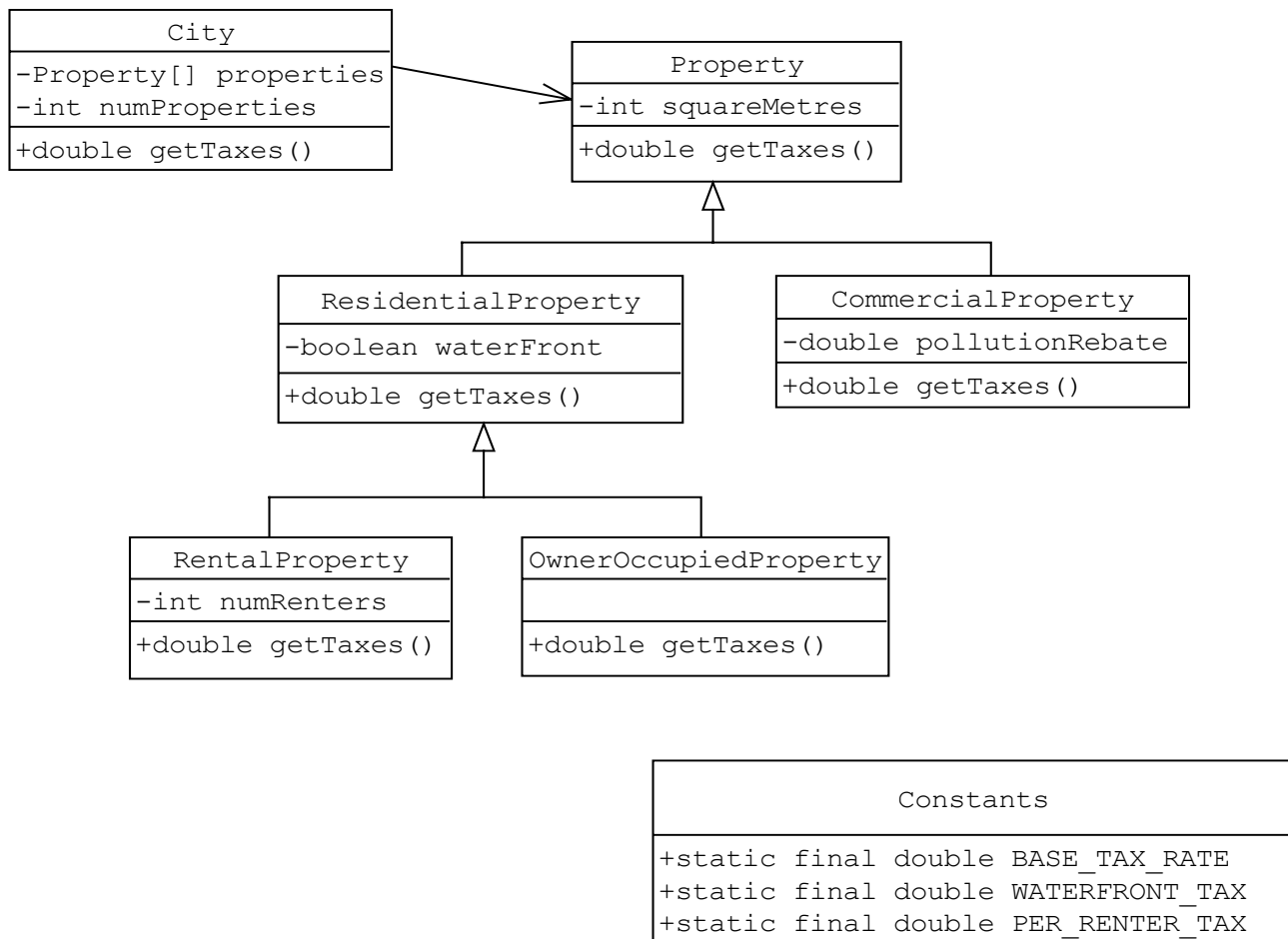


Name: _____

- b. [4] Show how these violations can be fixed by providing a new and improved UML class diagram for the `Course` class and any **new** classes.

Polymorphism [12 marks]

A city has designed classes to model properties as shown in the following UML class diagram. Note that the diagram is incomplete. To simplify the diagram, some instance variables and methods have been omitted.



The city needs to calculate the total property tax it should collect.

- The total property tax for the city consists of base property taxes assigned to every piece of property plus additional property taxes for some types of properties.
- The base property tax of a property is its size (in square metres) times `Constants.BASE_TAX_RATE`.
- Residential waterfront properties are charged an additional tax of `Constants.WATERFRONT_TAX`.
- Rental residential properties are charged an additional tax of `Constants.PER_RENTER_TAX` per renter.
- The tax for commercial properties is reduced by the pollution rebate assigned to them by environmental inspectors.

For each of the classes below, either

- implement the `getTaxes` method, or
- write “no implementation necessary; remove from UML class diagram”.

Name: _____

[2] in City:

```
public double getTaxes()  
{
```

[2] in Property:

```
public double getTaxes()  
{
```

[2] in ResidentialProperty:

```
public double getTaxes()  
{
```

[2] in RentalProperty:

```
public double getTaxes()  
{
```

[2] in OwnerOccupiedProperty:

```
public double getTaxes()  
{
```

[2] in CommercialProperty:

```
public double getTaxes()  
{
```

Name: _____

Class Discovery [5 marks]

The Sudsy-Wash Laundry Company is a business which operates several laundromats across Canada. Each laundromat contains coin-operated washing machines and dryers, which the company runs and maintains. The company would like to have a program written to model its business. Here are a few of the questions it is interested in:

- Which laundromats generate the most/least money for the company?
- How many loads of laundry are performed on different washing machines and dryers?
- Which washing machines use the most/least water?
- Which washing machines and dryers are out of order at any given time?
- How many dryers are owned by the company?

Identify four classes that would be essential when writing an object-oriented program to **model** this situation.

Provide the name of each class and a brief description of the class's purpose:

Class 1

| |
|--|
| |
|--|

Class 2

| |
|--|
| |
|--|

Class 3

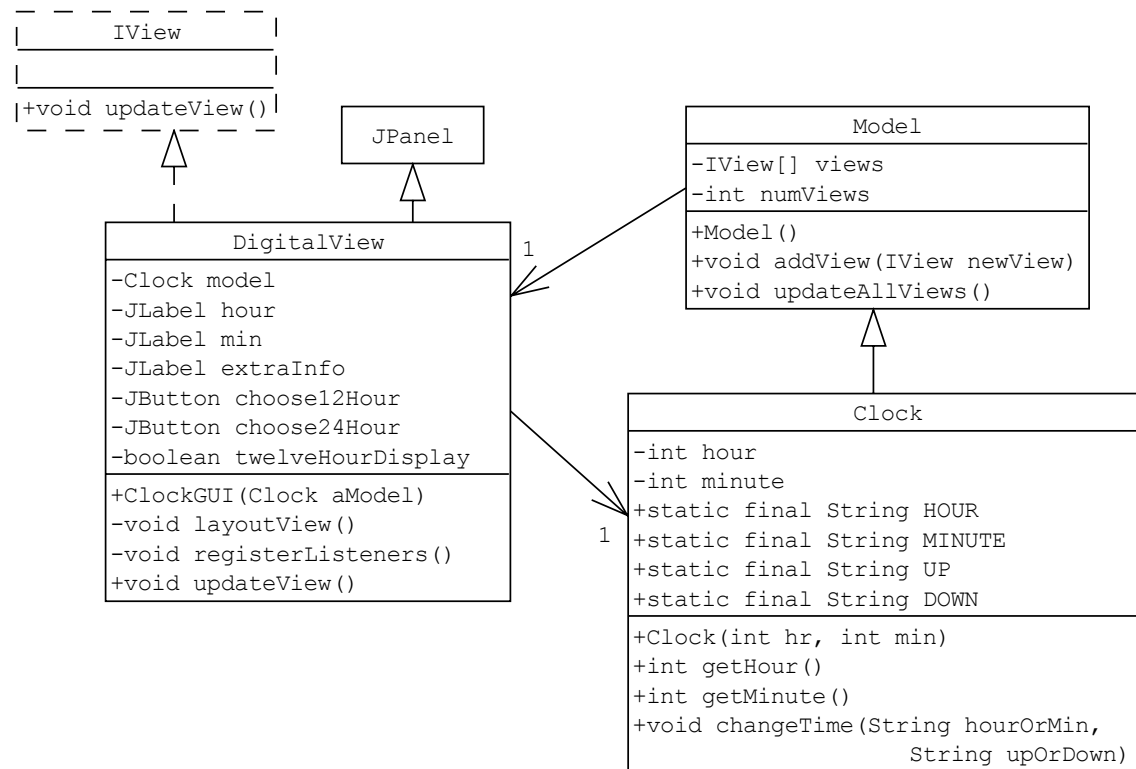
| |
|--|
| |
|--|

Class 4

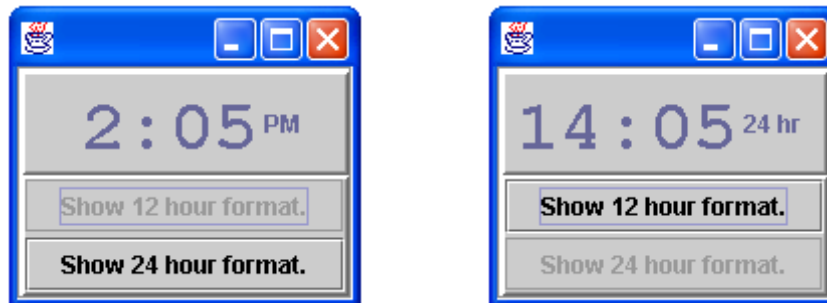
| |
|--|
| |
|--|

GUIs [30 marks]

Suppose you have the following classes representing a model and view of a simple digital clock:



The view allows the user the option to view the time in either a 12-hour format (showing AM or PM) or a 24-hour format. Here are a couple of examples of what the GUI could look like for the time 2:05 p.m.



Notes:

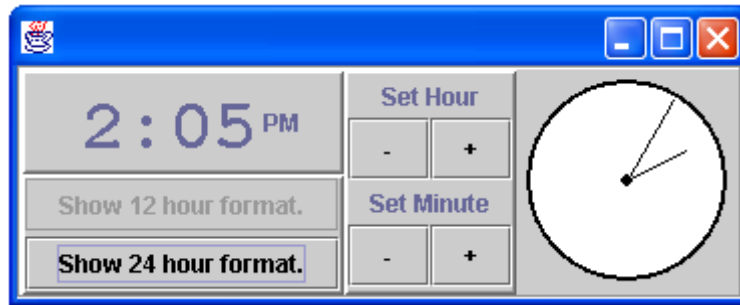
- The `JLabel` called `extraInfo` in `DigitalView` is the part of the GUI that shows one of “AM”, “PM”, or “24 hr”.
- The colon (`:`) is a `JLabel` which is constructed and added in the `layoutView` method of the `DigitalView` class. This `JLabel` is not an instance variable.
- `twelveHourDisplay` in `DigitalView` is `true` if and only if the clock should be displaying the time in a 12-hour format.
- A `Clock` object stores and **returns** the time in a 24-hour format. For example 0 hours and 0 minutes is midnight, 12 hours and 0 minutes is noon, and 20 hours and 15 minutes is equivalent to 8:15 p.m.
- The `changeTime` method will increase or decrease either the hour or the minute of the time based upon the values of the parameters given. Constants for `HOUR`, `MINUTE`, `UP`, and `DOWN` have been declared in the `Clock` class. For example, `changeTime(Clock.HOUR, Clock.UP)` increments the time by one hour.

Name: _____

- a. [9] Write the `updateView` method in the `DigitalView` class. The use of helper methods is encouraged. Assume the listeners for this view do not update any GUI components.

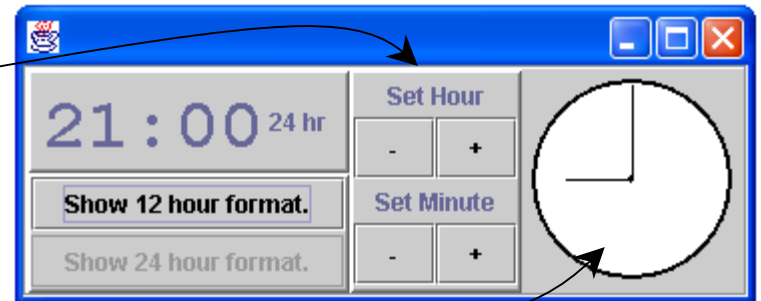
```
public void updateView()
{
```

Now suppose that you have a more complex GUI for this model that looks like this:



This GUI requires more classes in the view:

- ClockUI which holds the views
- SetTimeView which contains the instructions and buttons needed to change the time
- AnalogView which contains a picture of the clock



- b. [2] Complete the class declaration for the AnalogView.

```
public class AnalogView _____
```

- c. [2] Complete the class declaration for the SetTimeView.

```
public class SetTimeView _____
```

- d. [2] Other than “Start the program” and “Stop the program” identify two use cases for this clock program.

Name: _____

For the next two questions, assume that only the following instance variables have been declared in the `SetTimeView` class. You may not declare any additional instance variables.

```
private Clock model;  
private JButton hourUp = new JButton("+");  
private JButton hourDown = new JButton("-");  
private JButton minUp = new JButton("+");  
private JButton minDown = new JButton("-");
```

- e. [8] Complete the `registerListeners` method and any inner classes needed in the `SetTimeView` class. For full marks your solution should define and use just one listener. However, a correct solution that uses more listeners is worth more marks than an incorrect solution that uses fewer listeners. **Hint:** Remember that the `ActionEvent` class provides a method called `getSource`. This method returns a reference to the GUI component that triggered the event.

Complete the implementation of the `registerListeners` method below.

```
private void registerListeners()  
{
```

Define the necessary inner class(es) below.

Name: _____

- f. [7] Write the `layoutView` method for the `SetTimeView` class.

```
private void layoutView()
{
```

Arrays [12 marks]

Write a method to copy the border elements of a filled two-dimensional array of characters into a filled one-dimensional array of characters. Begin with the element in row 0 and column 0 and move “clockwise” ending with the element in row 1 and column 0. You may assume that there are at least two rows and two columns, and that the array is rectangular (that is, each row has the same number of columns).

For example, with input

```
Z D C D A F O
S I R K L M N
X A B C E G U
V F H W X B C
Q P N E F F L
Q D I C W L E
```

the array

```
Z D C D A F O N U C L E L W C I D Q Q V X S
```

should be returned.

```
public static char[] getArrayBorder(char[][] rectangle)
{
```

Multiple Choice [1 mark each]

For each question, circle the single best answer.

1. In CS 132, we used Java inner classes

- a) to allow static methods to access instance variables.
- b) to allow GUI listeners to access the instance variables of a view.
- c) to allow a model to call the `actionPerformed` method of a view.
- d) to allow `TextInput` to read data one line at a time.

2. Consider the following code fragment.

```
public static void main(String[] args)
{ String s = "hello";
  s[0] = 'j';
}
```

- a) The given code fragment will cause an error.
- b) The given code fragment will not cause an error.
- c) There is insufficient information to determine whether or not an error will occur.

3. Consider the following code fragment.

```
public static void main(String[] args)
{ String s = "hello";
  s = s + " world";
}
```

- a) The given code fragment will cause an error.
- b) The given code fragment will not cause an error.
- c) There is insufficient information to determine whether or not an error will occur.

4. Consider the following code fragment.

```
public static void main(String[] args)
{ String s = "hello";
  s.charAt(0) = 'j';
}
```

- a) The given code fragment will cause an error.
- b) The given code fragment will not cause an error.
- c) There is insufficient information to determine whether or not an error will occur.

5. Consider the following code fragment.

```
public static void main(String[] args)
{ String s = "hello";
  s = s + s.length();
}
```

- a) The given code fragment will cause an error.
- b) The given code fragment will not cause an error.
- c) There is insufficient information to determine whether or not an error will occur.

6. Walkthroughs are most closely related to

- a) polymorphism.
- b) stepwise refinement.
- c) use cases.
- d) UML class diagrams.

Each of the following code fragments is taken from either a well-designed final project or the Hangman example from practicum. In each case, indicate whether the code is best placed in one of the classes making up the *view* or best placed in one of the classes making up the *model*.

7. `this.repaint()`
 a) view
 b) model
8. `if (location.suitableForPlant())`
 `location.addPlant(newPlant)`
 a) view
 b) model
9. `this.updateAllViews()`
 a) view
 b) model
10. `addView(this)`
 a) view
 b) model
11. `this.setLayout(new GridLayout(26,2))`
 a) view
 b) model
12. `ButtonListener b = new ButtonListener(letter)`
 a) view
 b) model
13. `return plantLayer.initialCost() + plantLayer.maintenanceCost()`
 a) view
 b) model

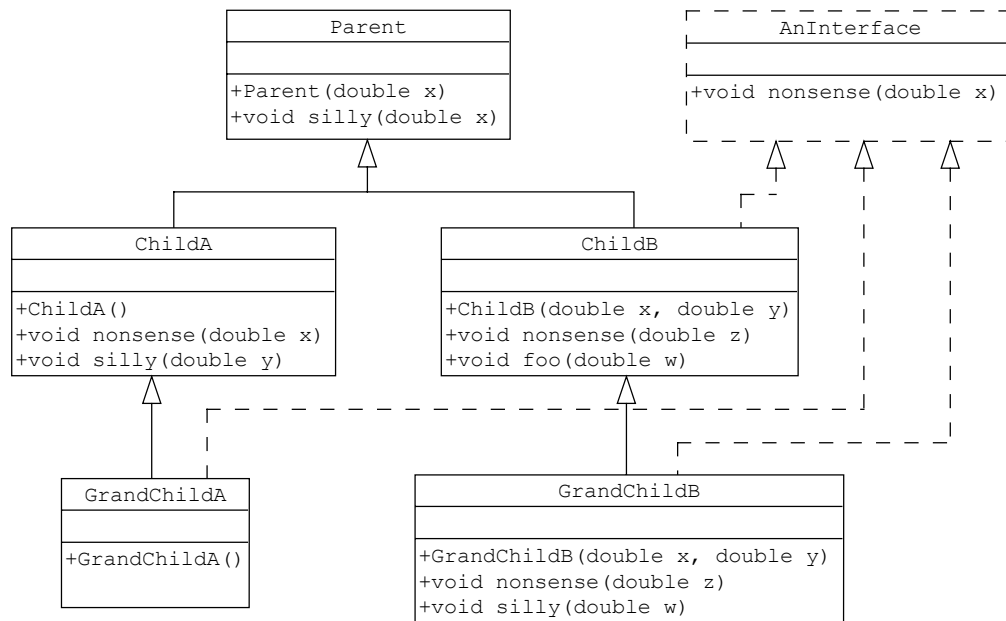
Types, Inheritance, Polymorphism and Interfaces [2 marks each]

The Reference Booklet contains details for the interface and classes needed to answer this question.

Consider each of the following code fragments. If execution of the code causes an error then indicate this by checking the box. If no error occurs then circle *all* the methods *and* constructors in the UML class diagram that are executed during execution of the code fragment.

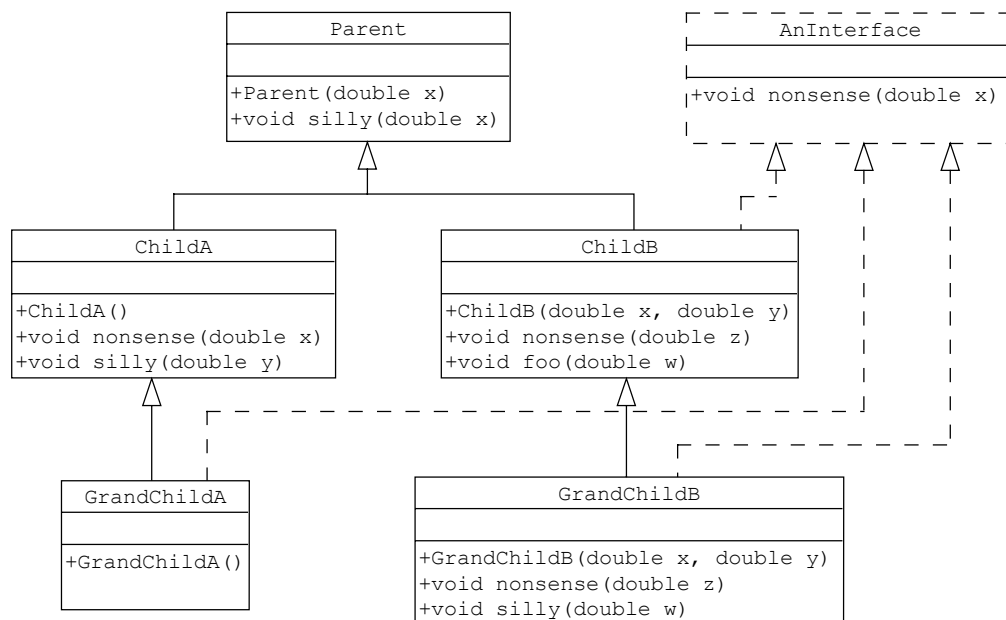
14. `Parent p = new ChildA();`
`p.silly(5.0);`

☐ ERROR



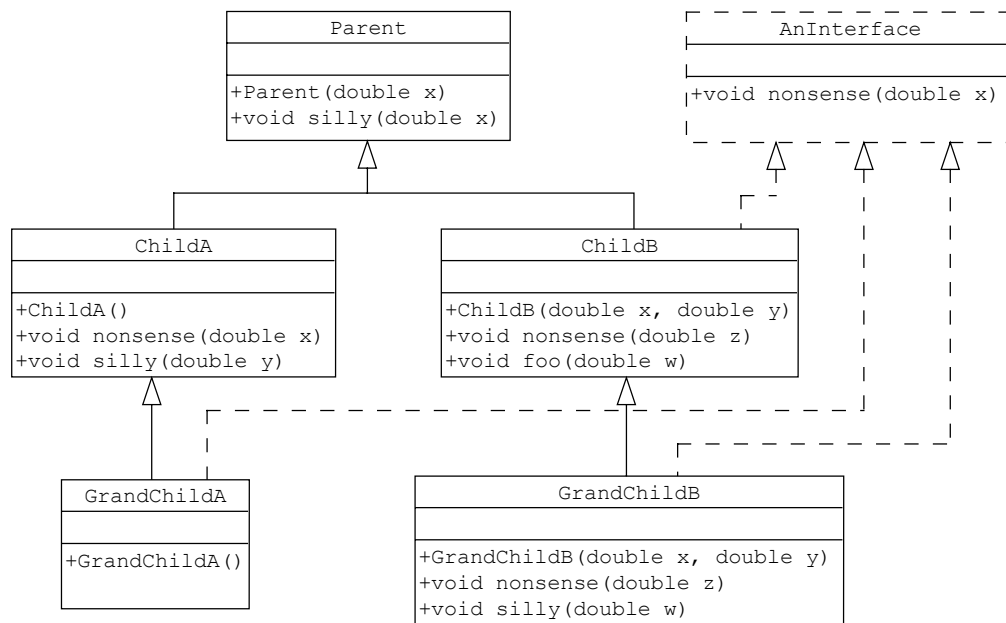
15. `Parent p = new ChildB(3.14, 2.718);`
`p.nonsense(1.0);`

☐ ERROR



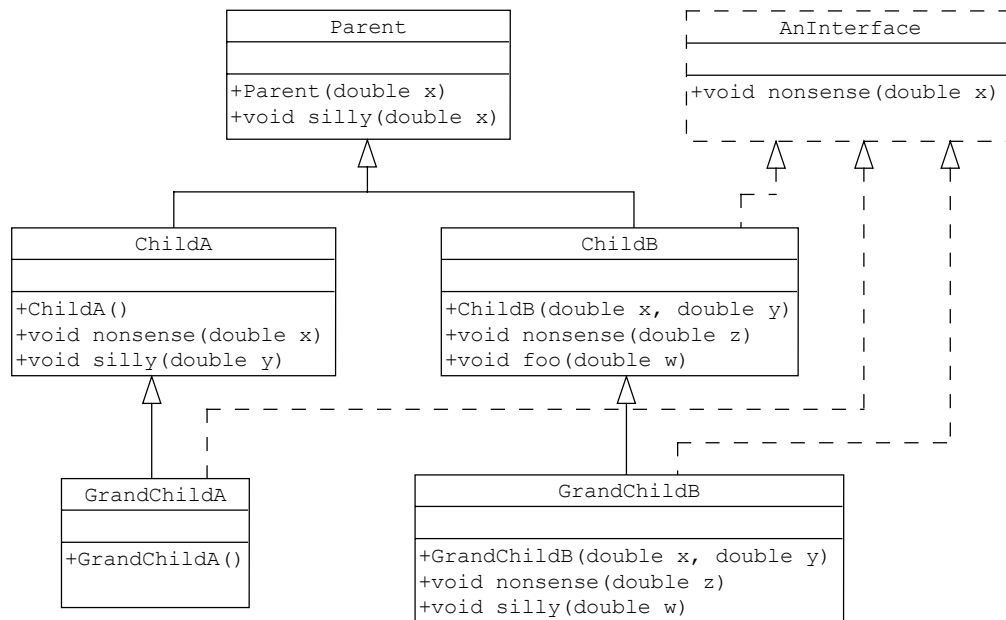
16. `ChildB c = new GrandChildB(4.4, 4.4);`
`c.foo(1.0);`

❑ ERROR



17. `AnInterface i = new GrandChildB(1.618, 1.618);`
`i.silly(3.7);`

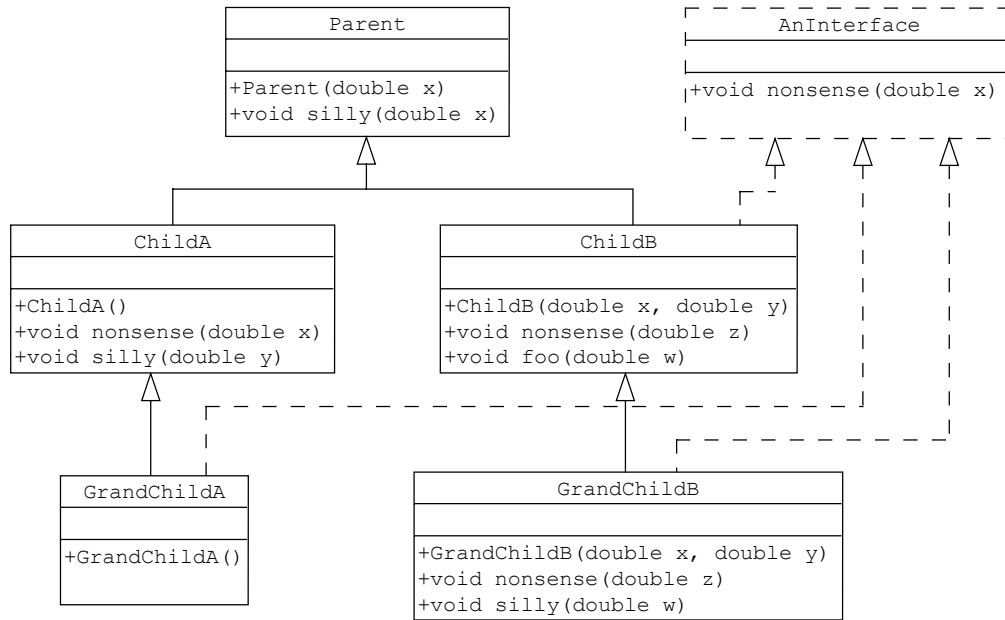
❑ ERROR



Name: _____

18. `AnInterface i = new GrandChildA();`
`i.nonsense(8.0);`

☐ ERROR



“Do You Really Get It?” [2 marks each]

Answer each of the following questions with a few sentences. Think carefully before you answer. In each case, simply stating things that are true is not sufficient. Strive to give the best, most meaningful response. Your answers should demonstrate that you have a deep understanding of the related course concepts.

19. When might it be better to use arrays instead of ArrayLists? Give two examples.

20. How can the use of polymorphism save programmers time?

21. Give a specific example illustrating that object-oriented heuristics can contradict each other.

22. Are there any disadvantages to separating the view from the model in a GUI-based program? Explain.

Name: _____

Extra Space

This space is available to continue answers started elsewhere in this booklet. Please label everything clearly, both here and where the question is asked so the marker will know where to look.