

EXAMINATION INSTRUCTIONS

This examination has 6 pages. Check that you have a complete paper.
Each candidate should be prepared to produce, upon request, his or her SUNY/UB card.

This examination has 5 questions. Answer all questions.

You have 60 minutes to complete this examination. Use your time accordingly.

READ AND OBSERVE THE FOLLOWING RULES:

- ▶ Names are pre-printed on the exam booklets. Ensure that you have YOUR exam.
- ▶ Sign, using your usual signature, in the space provided on the back cover.
- ▶ All of your writing must be handed in. This booklet must not be torn or mutilated in any way, and must not be taken from the examination room.
- ▶ Show all of your work in arriving at an answer, unless instructed otherwise. Partial credit will be awarded as appropriate.
- ▶ Candidates are not permitted to ask questions of the invigilators, except in cases of supposed errors or ambiguities in examination questions.
- ▶ CAUTION – Candidates guilty of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action.
 - ◆ Making use of any books, papers or memoranda, calculators or computers, audio or visual cassette players, or other memory aid devices, other than those explicitly authorised by the examiners.
 - ◆ Speaking or communicating with other candidates.
 - ◆ Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.

----- DO NOT WRITE BELOW THIS LINE! -----

Q1	Q2	Q3	Q4	Q5	TOTAL	%
/10	/10	/10	/10	/10	/50	/100

Question 1 [10 points, 2 points each]

The code given below is correct: it compiles without errors. I have added some extra spacing to make this question easier to answer.

Circle, and identify by number, one **and only one** example of each of the following items in the code below. If you believe no example exists, write “no example” next to that item in the list. To show you how I want the question answered, *the first one is done for you*.

- | | |
|---|---|
| 1. access control modifier | 4. looping statement (entire statement) |
| 2. accessor method definition | 5. the name of a supertype |
| 3. conditional statement (entire statement) | 6. an int literal |

```

package code.model;
import java.util.ArrayList;
import java.util.Random;
import code.ui.UI;

1 public class Model implements Observable {

    private UI _observer;
    private Random _rand;
    private ArrayList<String> _imageFileNames;
    private ArrayList<String> _spinnerCurrentValues;

    public Model() {
        _rand = new Random();
        _imageFileNames = new ArrayList<String>();
        _imageFileNames.add("Red.png");
        _spinnerCurrentValues = new ArrayList<String>();
        for(int i=0; i<_spinnerCurrentValues.size(); i=i+1) {
            _spinnerCurrentValues.add(i, _imageFileNames.get(_rand.nextInt(_imageFileNames.size())));
        }
    }

    public boolean jackpot() {
        for (int i=1; i<_spinnerCurrentValues.size(); i=i+1) {
            if ( ! _spinnerCurrentValues.get(i-1).equals(_spinnerCurrentValues.get(i)) ) {
                return false;
            }
        }
        return true;
    }

    public void addObserver(UI ui) {
        _observer = ui;
    }

    public String getImageFileName(int i) {
        return _spinnerCurrentValues.get(i);
    }
}

```

Question 2

☐ 10 points: perfect

☐ 7 points: essentially correct but with small mistakes

☐ 3 points: clearly wrong, some correct elements

☐ 0 points: for anything else

Study the following code, then answer the question which follows.

```
public String whatDoesThisDo(int x) {
    String result = "";
    if (x < 55) {
        return "Zoo";
    }
    else if (x < 70) {
        result = "Walk";
        x = x - 55;
    }
    else if (x < 85) {
        result = "Talk";
        x = x - 70;
    }
    else {
        result = "Hawk";
        x = x - 85;
    }
    if (x < 5) {
        result = result + "??";
    }
    else if (x < 10) {
        result = result + "!!";
    }
    else if (x < 15) {
        result = result + "***";
    }
    return result;
}
```

[5 points] What does the following statement print?

```
System.out.println("Answer is " + whatDoesThisDo(71));
```

Write your answer below:

[5 points] Give a value for `x` such that `whatDoesThisDo(x)` returns "Hawk***"?

Write your answer below:

Question 3

☐ 10 points: perfect

☐ 7 points: essentially correct but with small mistakes

☐ 3 points: clearly wrong, some correct elements

☐ 0 points: for anything else

Define a method which takes two parameters, an `ArrayList<String>` and an `int`, which prints (using `System.out.println`) all the `Strings` from the `ArrayList` whose length is greater than or equal to the `int`.

For example, if the method is named `printer` and is defined in a class named `Question3`, then

```
new Question3().printer(null, 4)
```

must not produce any runtime errors and must print nothing, whereas

```
ArrayList<String> list = new ArrayList<String>();
list.add("foo");
list.add("fluffy");
list.add("pi");
list.add(null);
list.add("cake");
list.add("rho");
new Question3().printer(list, 4);
```

must not produce any runtime errors and must produce the following output:

```
fluffy
cake
```

Write your answer below:

Question 4

☐ 10 points: perfect

☐ 7 points: essentially correct but with small mistakes

☐ 3 points: clearly wrong, some correct elements

☐ 0 points: for anything else

Study the following code:

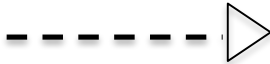
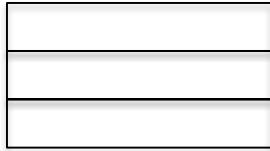
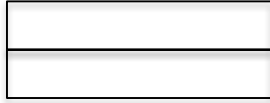
```
public void mystery(String s, int w) {  
    for (int i=0; i<w; i=i+1) {  
        System.out.print(s);  
        for (int k=0; k<i; k=k+1) {  
            System.out.print(".");  
        }  
        System.out.println(s);  
    }  
}
```

Show what is printed by the following method call:

```
mystery(" ", 4)
```

Question 5 [10 points – 2 points each]

For each UML symbol below, match it with its correct interpretation from the list given on the right (draw a line from the symbol to the correct text). Since there are more descriptions than symbols some of the descriptions will be unused.



1. inheritance relationship
2. variable
3. realization (implementation) relationship
4. package
5. composition relationship
6. method
7. association relationship
8. class box
9. dependency relationship
10. interface box