1006 Midterm Solutions

Question 1

When an object is created with `new` in Java, where are the object's contents (such as its attributes) stored [**1 mark**]? What exactly is stored in a variable when the result of `new` is assigned to it [**1 mark**]?

Expected answer

The object's contents are stored in the heap. A reference to the location on the heap is stored in the variable.

Question 2

What is method overloading [**1 mark**] and when is it useful [**1 mark**]?

Expected Answer

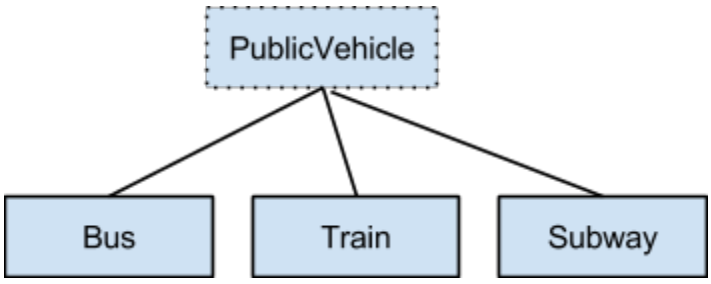
Method overloading is when you write multiple methods with the same name but different parameters. It is useful when you want to allow users of your method to call it with multiple different types of objects, while still using a single method name (makes it easier to use).

Question 3

Suppose you wanted a class hierarchy to represent various public transportation vehicles. Draw a possible hierarchy that includes at least three concrete classes shown with solid lines [**0.5 marks**] and at least one abstract class shown with dotted lines [**0.5 marks**]. Explain why you organized your hierarchy this way [**1 mark**].

Expected Answer

Here is an example (there are many possibilities):



I made the PublicVehicle class abstract because it is acting as an organizing class, but isn't something that should exist as an object. The Bus, Train, and Subway classes represent specific kinds of public vehicles, and are the actual "things" that should be able to exist.

Question 4

Explain how Java interfaces work [**1 mark**] and give one example of why/when you might use them [**1 mark**].

Expected Answer

Java interfaces are a collection of methods that all classes implementing the interface must write. They are useful when you want to ensure an object has a certain set of behaviours, or they are useful to relate objects that have something in common but aren't an "is a" relationship (e.g. for polymorphism).

Question 5

What will the code at right print out when compiled and run [**1 mark**]? Why [**1 mark**]?

Expected Answer

Output is:

```
a test.
is
This
```

In the main method, a new Test object is created with a constructor that takes an int. In that constructor, a super-constructor that takes a string is called. Inside SuperTest, test() is called, printing out "a test." Then s (specifically, "is") is printed, and finally "This" is printed.

Question 6

Will the code at right compile and/or run [**1 mark**]? Why or why not [**1 mark**]?

Expected Answer

The code WILL compile. There will be a runtime error since an object that is really of type A can't be cast to an object of type B (since B is a subclass of A). It is not a compile error because Java doesn't know ahead of time what the object being cast will actually be. Removing the line with the cast will cause the code to run well.

Question 7

Part 1

Write a proper **Book** constructor that receives the following 3 parameters: a string for the title of the book, an array of strings as the list of authors, and the year that the book was printed [**1 mark**].

Expected Answer

```
public Book(String x, String[] y, int z)
{
      title = x;
      authors = y;
      year = z;
}
```

Part 2

In the **Library** class, complete the method called **addBook** that receives a **Book** object as a parameter, and then adds it to the **books** array [**1 mark**]. If the array is full, do not add the book [**1 mark**].

```
public void addBook(Book b)
{
      if (numberOfBooksSoFar < books.length)
      {
        books[numberOfBooksSoFar] = b;
        numberOfBooksSoFar++;
      }
}
```

Part 3

Briefly explain how we might make use of polymorphism if we were to add different materials (in addition to books) to the library [**1 mark**].

Expected Answer

Have a superclass or interface that organizes the different materials into a hierarchy (something like “BorrowableMaterial”). Then we can store all the different types of materials into an array of that type, but still use the common functions of the materials. When such methods are called, the actual object’s methods are used instead of the superclass’s.

Question 8

Part 1

Write the following method so that it deposits the given amount to the account properly and creates and stores a proper transaction in the transactions array [**2 marks**]. You may assume there will always be room in the array.

(Should have read “Write the following method so that it represents the bookstore selling a book…” - marks can be given if there was confusion based on the editing error.)

Expected Answer

```
transactions[transactionCount] = new SellBookTransaction(amount);
transactionCount = transactionCount + 1;
```

Part 2

Complete the following method that displays a report showing the total of all withdrawals that have been made from the account as well as the total of all deposits made as follows [**2 marks**]:

```
Total of sold:        $452.87    // don't worry about proper
formatting
Total of purchased:  $1918.12    // don't worry about proper
formatting
```

Expected Answer

```
float totalSold = 0;
float totalBought = 0;

for (int i=0; i < transactions.length; i++)
{
  Transaction t = transactions[i];
  if (t instanceof SellBookTransaction)
    totalSold += t.amount;
  else
    totalBought += t.amount;
}

System.out.println("Total of sold: $" + totalSold);
System.out.println("Total of purchased: $" + totalBought);
```

Again mixing up sell/buy (reversing them) is ok.