Homework 03 – Michael Ng

2.2

You get 0.

2.3

The machine can print ticket no matter how much balance is in the machine (too much or below the price) and will always reset the balance to 0 after printing the ticket. No refunds are given.

2.5

Yes, the ticket cost is a different number when printed on the second machine.

2.6

public class Student

{

// Code omitted

}

public class LabClass

{

// Code omitted

}

2.7

Yes. “class public TicketMachine” produces a compile error (<identifier> expected), and “public class TicketMachine” does not. The error “<identifier> expected” does not clearly explain what is wrong.

2.8

Yes, it is possible to leave out the keyword “public” in the TicketMachine class. It does not cause any compile errors or runtime errors.

2.10

private int price;

private int balance;

private int total;

public TicketMachine(int cost){}

public int getPrice(){}

public int getBalance(){}

public void insertMoney(int amount){}

public void printTicket(){}

2.11

The constructor defines or sets the fields and does not have the keyword “int” or “void” in the header.

2.12

1. Integer
2. Student
3. Server

2.13

1. alive
2. tutor
3. game

2.14

Representative, host, tutor, and game are most likely to be class names.

2.15

Yes, other orderings will result in an error. “private int price” is the only order that does not result in an error.

2.16

Yes, otherwise there will be a syntax error.

2.17

private int status;

2.18

The Student class.

2.19

The constructor has two parameters, one String type and one double type.

2.20

There may be a String type field and double type field, looking at the parameter.

2.21

name = petsName;

2.22

public Date(String month, int day, int year){}

2.23

Both methods are very similar, but they both return different values when called.

2.24

“What is my current balance?”

2.25

No, the accessor method’s name is independent of the field’s name.

2.26

public int getTotal()

{

return total;

}

2.27

A “missing return statement” error

2.28

The method getPrice() has a return statement, while printTicket() does not.

2.29

No, they are mutator methods, which update the value of fields (they also don’t need return methods). Because of the keyword ‘void’ in the method header, the method does not return a value.

|2.30|

2.31

There is the keyword “void”. Constructor methods do not have a keyword such as int or void.

2.32

public void setPrice(int cost)

{

price = cost;

}

2.33

public void increase(int points)

{

score = score + points;

//score += points;

}

2.34

Yes, this is a mutator method because it changes the value of score based on user input. When calling the method ‘increase’, typing an integer to the parameter will increase score by that integer.

2.35

public void discount(int amount)

{

price = price – amount;

//price -= amount;

}

2.36

My cat has green eyes.

2.37

public void prompt()

{

System.out.println("Please insert the correct amount of money.");

}

2.38

The statement would print out “# price cents.” (quotation marks not included)

2.39

It would also print “# price cents.”

2.40

No, both statements do not refer to ‘price’ in the TicketMachine class and will not return the actual price.

2.41

public void showPrice()

{

System.out.println("The price of a ticket is " + price + " cents.");

}

2.42

Both have different outputs. This is because the field ‘price’ in ticketMachine1 holds a different value than the field ‘price’ in ticketMachine2.

2.43

Modified:

public TicketMachine()

{

price = 1000;

balance = 0;

total = 0;

}

The modified version does not ask the user to input a price for the ticket, making the default price of a ticket 1,000 cents.

2.44

Added:

public TicketMachine(int newPrice)

{

price = newPrice;

balance = 0;

total = 0;

}

2.45

public void empty()

{

total = 0;

}

This method doesn’t need a parameter since it sets total to 0. The empty method is a mutator because it changes the value of total.