A Class in Java/Processing is a _____.

_____ have _____ and _____ about an _____, but it is NOT the _____. People use _____ as a _____ to build _____.

A class is a _____ for the _____. It has _____ and _____ that the computer uses to build objects. The class is NOT the _____ itself.

A class in Java has _____ parts:

a)

b)

c)

The following is an example of a class (label the parts):

```
class Dice  {

  int sideUp;
  int numSides;

  Dice()  {
    numSides = 6;
    sideUp = roll();
  }

  void roll()  {
    sideUp = (int)(Math.random() * numSides);
  }

  int getSideUp()  {
    return sideUp;
  }
}
```

How do we get the computer to use a class to create an object? _____

How do we get the object to do things? _____

Label the three parts of a class on the example class below:

```
class Integer {
  int val;

  Integer(int v)  {
    val = v;
  }
  int getValue()  {
    return val;
  }
}
```

What two parts of a class have the same name?

Write the code that would instantiate an `Integer` object:

Write the code that would get the value from the object you created in the question above:

Write the `Wall` class for our Angry Birds clone. The `Wall` class will have five facts: x-coordinate, y-coordinate, width, and height, and color. It will have one constructor with 4 parameters that will set the `x`, `y`, `width` and `height` parameters. By default, the color of `Wall` objects will be `green`. The `Wall` class has 2 methods – `display()` and `move()`. The `display()` method will draw the appropriate color rectangle (with black border) at the appropriate spot. The `move()` method will cause the rectangle to move to the left by 5 pixels. Use your notes to help write the class on the back of this page.

Key

Class Worksheet 1 | 2014

Label the three parts of a class on the example class below:

```
class Integer {
  int val;

  Integer(int v) {
    val = v;
  }
  int getValue() {
    return val;
  }
}
```

IV

constructor

method

What two parts of a class have the same name? Name + constructor

Write the code that would instantiate an `Integer` object:

```
Integer bob = new Integer(6);
```

one ✓ if no #

fun ✓ s if # inside

Write the code that would get the value from the object you created in the question above:

```
bob.getValue()
```

Write the `Wall` class for our Angry Birds clone. The `Wall` class will have five facts: x-coordinate, y-coordinate, width, and height, and color. It will have one constructor with 4 parameters that will set the `x`, `y`, `width` and `height` parameters. By default, the color of `Wall` objects will be `green`. The `Wall` class has 2 methods – `display()` and `move()`. The `display()` method will draw the appropriate color rectangle (with black border) at the appropriate spot. The `move()` method will cause the rectangle to move to the left by 5 pixels. Use your notes to help write the class on the back of this page.

```
class Wall ✓
{
  float x;
  float y;
  float w;
  float h;
  color c;

  Wall ( float xOne, float yOne, float wd, float ht)
  {
    x = xOne
    y = yOne ✓
    w = wd;
    h = ht;
    c = color (0, 255, 0);
  }

  void display()
  {
    stroke(0)
    fill(c)
    rect(x, y, w, h)
  }

  void move()
  {
    x = x + 5;
  }
}
```

5 IVs

Write the `Mario` class for Nintendo. The `Mario` class will keep track of the number of lives, number of coins, and whether or not he is big. By default, a `Mario` object will have `0` coins, `3` lives, and he will not be big. The `Mario` class has 2 methods – `addCoin()` and `changeSize()`. The `addCoin()` method will take `Mario's` coin count and add one. `addCoin()` will also reset the coin count to `0` and add one to the number of lives if the coin count reaches `100` coins. The `changeSize()` method will "flip" whether or not the `Mario` object is big or not. If a `Mario` object is NOT big and `changeSize()` is called then the `Mario` object will become big. The opposite is true the next time `changeSize()` is called. Use the space below to write the class.

Given the following class:

```
class Student {
  String name;
  int id;
  float gpa;

  Student(String n, int i)  {
    name = n;
    id = i;
    gpa = 0;
  }

  void setGPA(float newGpa)  {
      gpa = newGpa;
  }
  int getID()  {
      return id;
  }
}
```

How many instance variables are there in the `Student` class? _____

How many constructors are there in the `Student` class? _____

How many methods are there in the `Student` class? _____

Write the code that would instantiate a `Student` object:

Assume there is a `Student` object named `stu`, write the code to change `stu`'s gpa to 3.5:

Assume there is a `Student` object named `stu`, write the code to print `stu`'s id:

What are the two differences between a Constructor and other methods:

1)
2)

A class is also referred to as a _____

Write a method called `changeID()` that will be added to the `Student` class. This method, when called, will update a `Student`'s `id` to the parameter that was sent to it. For example, `stu.changeID(23456)`, would change `stu`'s id to 23456.

Key

Class Quiz 1 | 2014

Given the following class:

```
class Student {
  String name;
  int id;
  float gpa;

  Student(String n, int i)  {
    name = n;
    id = i;
    gpa = 0;
  }

  void setGPA(float newGpa)  {
      gpa = newGpa;
  }
  int getID()  {
      return id;
  }
}
```

IV

1 Constr.

2 M

How many instance variables are there in the Student class? 3

How many constructors are there in the Student class? 1

How many methods are there in the Student class? 2

Write the code that would instantiate a Student object:

```
Student s = new Student("bob", 7);
```

Assume there is a Student object named stu, write the code to change stu's gpa to 3.5:

```
stu.setGPA(3.5);
```

Assume there is a Student object named stu, write the code to print stu's id:

```
print(stu.getID());
```

What are the two differences between a Constructor and other methods:

1) no return type
2) same name as class

A class is also referred to as a blueprint

Write a method called changeID() that will be added to the Student class. This method, when called, will update a Student's id to the parameter that was sent to it. For example, stu.changeID(23456), would change stu's id to 23456.

```
void changeID(int other)
{
    id = other;
}
```

You have been hired by Taito to help write their new game called Space Invaders. You are in charge of writing the `Ship` class. The ship in space invaders has an `x,y-coordinate` position as well as a variable to keep track of the number of lives. There are two ways to instantiate a `Ship` object. By default, it will start with 5 lives as well as be set to the position `(400, 600)`. You can also instantiate a `Ship` object using its one parameter constructor that allows you to start the `Ship` at `(400,600)` but you can choose how many lives you have. The `Ship` class has two methods. The `display()` method will draw a white rectangle that is `100` pixels wide and `50` pixels tall whose center is the (x,y) coordinate of the `Ship` object. The other method called `fire()` will instantiate a `Bullet` object using the `Bullet` constructor below:

```
Bullet(float speed)
{
//body not shown
}
```

Then add that `Bullet` object to a `List` object called `bList`. The `List` class has one method called `add(Bullet name)` that will put the `Bullet` object in the list.

```
class Ship {
  float x;
  float y;
  int lives;

  Ship() {
    x=400;
    y=600;
    lives =5;
  }

  Ship(int lv) {
    x=400;
    y=600;
   lives =lv;
  }

  void display(){
    rectMode(CENTER);
    fill(255);
    rect(x,y,100,50);
  }

  void fire(){
    Bullet b = new Bullet(4.5);
    bList.add(b);
  }
}
```

You have been hired by Taito to help write their new game called Space Invaders. You are in charge of writing the `Ship` class. The ship in space invaders has an `x,y-coordinate` position as well as a variable to keep track of the number of lives. There are two ways to instantiate a `Ship` object. By default, it will start with 5 lives as well as be set to the position `(400, 600)`. You can also instantiate a `Ship` object using its one parameter constructor that allows you to start the `Ship` at `(400,600)` but you can choose how many lives you have. The `Ship` class has two methods. The `display()` method will draw a white rectangle that is `100` pixels wide and `50` pixels tall whose center is the (x,y) coordinate of the `Ship` object. The other method called `fire()` will instantiate a `Bullet` object using the `Bullet` constructor below:

```
Bullet(float speed)
{
//body not shown
}
```

Then add that `Bullet` object to a `List` object called `bList`. The `List` class has one method called `add(Bullet name)` that will put the `Bullet` object in the list.

Key

Class Quiz 2 – Write a class | 2014

You have been hired by Taito to help write their new game called Space Invaders. You are in charge of writing the `Ship` class. The ship in space invaders has an `x,y-coordinate` position as well as a variable to keep track of the number of lives. There are two ways to instantiate a `Ship` object. By default, it will start with `5` lives as well as be set to the position `(400, 600)`. You can also instantiate a `Ship` object using its one parameter constructor that allows you to start the `Ship` at `(400,600)` but you can choose how many lives you have. The `Ship` class has two methods. The `display()` method will draw a white rectangle that is `100` pixels wide and `50` pixels tall whose center is the (x,y) coordinate of the `Ship` object. The other method called `fire()` will instantiate a `Bullet` object using the `Bullet` constructor below:

```
Bullet(float speed)
{
//body not shown
}
```

Then add that `Bullet` object to a `List` object called `bList`. The `List` class has one method called `add(Bullet name)` that will put the `Bullet` object in the list.

```
class Ship {
  float x;
  float y;
  int lives;

  Ship() {
    x=400;
    y=600;
    lives =5;
  }

  Ship(int lv) {
    x=400;
    y=600;
   lives =lv;
  }

  void display(){
    rectMode(CENTER);
    fill(255);
    rect(x,y,100,50);
  }

  void fire(){
    Bullet b = new Bullet(4.5);
    bList.add(b);
  }
}
```

heading ✓ ✓ ✓ ✓ ✓ attempt @ IV

heading ✓ ✓ ✓ ✓ ✓ attempt to fill IV

heading ✓ ✓ ✓ ✓ ✓

heading ✓ ✓ ✓ ✓ ✓ ✓

5 pts/✓

Class and Loop Test

True/False
Indicate whether the statement is true or false.

____ 1. A class is often called a blueprint.

____ 2. A class has two parts: instance variables and constructors

____ 3. A constructor has the same name as the class and has a return type of `void`.

____ 4. The purpose of a constructor is to fill the instance variables.

Multiple Choice
Identify the choice that best completes the statement or answers the question.

```
class Donut {
  float price;
  String name;
  boolean hasSprinkles;

  Donut() {
    price = 0.65;
    name = "plain"
    hasSprinkles = false;
  }

  Donut(float p, String s, boolean yummy) {
    price = p;
    name = s;
    hasSprinkles = yummy;
  }

  float getPrice() {
    return price;
  }
}
```

____ 5. Which of the following is a valid instantiation of the `Donut` class?

a. `Donut bob = new Donut;`
b. `Donut bob = new Donut();`
c. `Dount bob = new Donut(price,name,hasSprinkles);`
d. `Donut bob = new Donut(0.99, "White Beauty", true);`
e. B and C are correct
f. B and D are correct
g. C and D are correct

____ 6. Assume that there is a `Donut` object called `mine`. Which of the following lines of code would print `mine`'s price on the screen?

a. `Donut.getPrice();`
b. `println(Donut.getPrice());`
c. `mine.getPrice();`
d. `println(mine.getPrice());`
e. `println(getPrice());`
f. `println(mine.getPrice);`
g. `mine.price`

```
class Player {
  int hits;
  int atBats;
  String name;

  Player(int h, int a, String n) {
   hits = h;
   atBats = a;
   name = n;
  }

  float getAverage() {
    <*1>
  }

  void addHit()
  {
    <*2>
  }
}
```

____ 7. Write the code to replace <*1> so that the `getAverage()` method returns the proper batting average for the a `Player` object. Batting average is calculated by diving number of hits by total at bats.

a. `return hits/atBats;`
b. `return 1.0 * hits/atBats;`
c. `1.0 * hits/atBats;`
d. `print( hits/atBats);`
e. `print(1.0 * hits/atBats);`

____ 8. Write the code to replace <*2> so that the `addHit()` method adds one to the total number of hits a `Player` object has.

a. `atBats++;`
b. `hits++;`
c. `hits = hits + 1;`
d. `h++;`
e. `hits + 1;`
f. more than one answer is correct.

____ 9. Which of the following is a valid instantiation of the `Player` class?

a. `Player bob = new Player();`

b. `Player bob = new Player(100.0,345,"Bob Jeroo");`
c. `Player bob = new Player(h,a,n);`
d. `Player bob = new Player(50, 100, "awesome sauce");`
e. B and C are correct
f. B and D are correct
g. C and D are correct

Numeric Response

```
class Donut {
  float price;
  String name;
  boolean hasSprinkles;

  Donut() {
    price = 0.65;
    name = "plain"
    hasSprinkles = false;
  }

  Donut(float p, String s, boolean yummy) {
    price = p;
    name = s;
    hasSprinkles = yummy;
  }

  float getPrice() {
    return price;
  }
}
```

10. How many instance variables does the `Donut` class have?

11. How many constructors does the `Donut` class have?

12. How many methods does the `Donut` class have?

```
class Player {
  int hits;
  int atBats;
  String name;

  Player(int h, int a, String n) {
   hits = h;
   atBats = a;
   name = n;
  }

  float getAverage() {
    <*1>
  }
```

```
  void addHit()
  {
    <*2>
  }
}
```

13. How many instance variables does the `Player` class have?

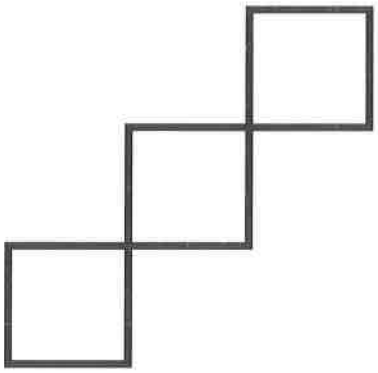
14. How many constructors does the `Player` class have?

15. How many methods does the `Player` class have?

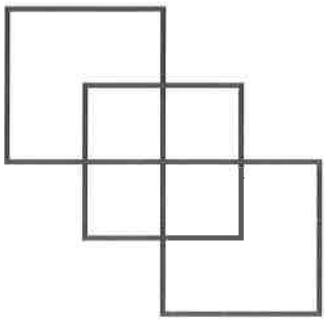
Matching

Match the following picture with the appropriate for loop below. **NOT ALL ANSWERS WILL BE USED **

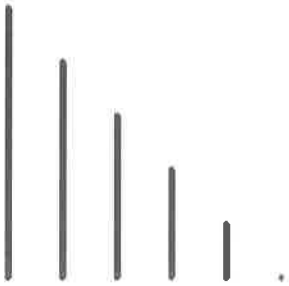
a.



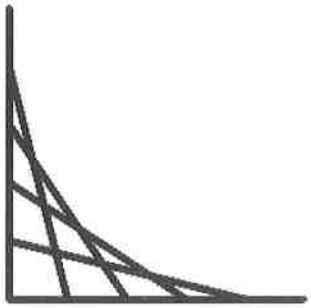
b.



c.



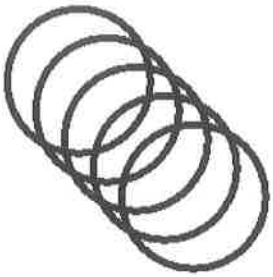
f.



g.

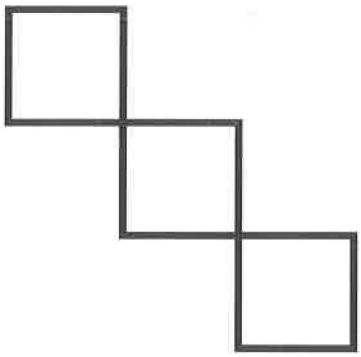
h.

d.



i.

e.



____ 16.
```
for(int i=0; i< 3; i++)
  rect(400-i*80, i*80, 80,80);
```

____ 17.
```
for(int i=0; i< 3; i++)
  rect(i*80, i*80, 160,160);
```

____ 18.
```
for(int val = 0; val <= 200; val += 40)

line(0,val,val,200);
```

____ 19.
```
for(int siz = 20; siz < 120; siz += 20)

   ellipse(200,200,siz,siz);
```

____ 20.
```
for(int myst = 50; myst >= 10; myst -= 10)
  rect(50-myst,0,myst,myst);
```

Class and Loop Test
Answer Section

TRUE/FALSE

1. ANS: T PTS: 1
2. ANS: F PTS: 1
3. ANS: F PTS: 1
4. ANS: T PTS: 1

MULTIPLE CHOICE

5. ANS: F PTS: 1
6. ANS: D PTS: 1
7. ANS: B PTS: 1
8. ANS: F PTS: 1
9. ANS: D
   ```
   Player(int h, int a, String n)
   ```

   PTS: 1

NUMERIC RESPONSE

10. ANS: 3

    PTS: 1
11. ANS: 2

    PTS: 1
12. ANS: 1

    PTS: 1
13. ANS: 3

    PTS: 1
14. ANS: 1

    PTS: 1
15. ANS: 2

    PTS: 1

MATCHING

16. ANS: A PTS: 1
17. ANS: B PTS: 1

18. ANS: F PTS: 1
19. ANS: G PTS: 1
20. ANS: H PTS: 1