INDEX

Five Environment Layout

The learning outcomes of Narrate are based on the C++ Early Objects ninth edition introductory programming textbook by Tony Gaddis, Judy Walters, and Godfrey Muganda. Narrate is composed of five different environments, each of which has its own set of learning outcomes. This section will detail the learning outcomes and narrative structure of each environment.

Environment 1

Learning Outcomes: Programming Basics

After completing the first environment the user should be able to :

1. Understand what a variable is
2. Understand that there are different data types
3. Understand that variables can be assigned values
4. Understand variable declaration
5. Recognize relational operators
6. Understand conditional statements
7. Understand branching
8. Understand looping
9. Recognize functions

Narrator:

Welcome to Narrate! I'm so excited to embark on this new journey with you!

Before we begin, I need to ask you a few questions:

Pretest question #1:

What is a variable?

Something that stores information. It can be changed.

Something that stores information. It cannot be changed.

An object defined by the programmer

Pretest question #2:

Which of the following is the data type: word draw ()

word

draw

()

Pretest question #3:

What does the following piece of code do: age = 7

it checks to see if age is equal to 7

it assigns age the value of 7

it assigns 7 the value of age

Pretest question #4:

Which of the following is a variable declaration without a variable initialization?

number age = 7

age = 7

number age

none of the above

Pretest question #5:

What does the relational operator do?

It creates a relationship between two things

It tests the relationship between two things

It eliminates the relationship between two things

Pretest question #6:

What is a conditional statement?

It allows a piece of code to execute only if a certain condition is met

It allows a piece of code to execute regardless if a certain condition is met

It causes the computer to turn off if a certain condition is met

Pretest question #7:

What is branching?

It allows certain pieces of code to execute depending on if the condition is met

It allows every piece of code to execute regardless if the condition met

It allows certain pieces of code to execute while a condition is met

Pretest question #8:

What is looping?

It allows every piece of code to execute regardless if the condition met

It allows code to execute while a condition is met

It allows code to execute once if a condition is met

Pretest question #9:

Which of the following is a function?

number age = 8

word name = “Sarah”

void set_name ()

number get_age ()

two of the above

A variable is like a box that holds information. *(T1.1)*

Variables can be assigned values. *(T1.3)*

Let’s define and assign values to some variables!

Enter your information in the underlined areas.

word name = “_____”

word favorite_color = “_____”

number age = _____

(A1.1, A1.3)

Notice that there are different types of variables! The **bolded** terms are **data types.**

Data is basically information

Data type defines the type of information that can be held

For instance, **word** name means that the variable name can only hold words

number age means that the variable age can only hold numbers

(T1.2)

What does **food** breakfast mean?

That the variable breakfast can only hold foods

That the variable breakfast can hold foods or data similar to foods like drinks

(A1.2)

Perfect! Now the variable name stores: some_name

the variable favorite_color stores: some color

and the variable age stores some_age

Is this information correct? Select your answer!

word correct = “no”

word correct = “yes”

(T1.3)

Great! Now let's complete your little brother's information! Initialize his name!

number brother_age = 10

word name = "_____"

(A1.1, A1.3)

Perfect! You're little brother goes to the school next to yours so your mom will be taking you guys to school together every morning.

Great! Now that you have entered your information let's start the story!

It's your first day of middle school at your new school. Are you feeling excited?

Notice in this example that first we declare "**word** excited" without **initializing** it with a value.

This is called variable declaration

(T1.4)

emotion excited

excited = "yes"

excited = "no"

(A1.1, A1.2, A1.4)

Well I'm sure you will have a wonderful year!

Mom:

BrotherName and UserName time to go to school, come on!....

Here we are! UserName don't forget to stop by the principal's office before the first period….

Have a great day! I love you!...

User:

Wow… this school is huge… And some of these people look like actual adults!...

Narrator:

Let's go to the principal's office for the new student welcome

Principal:

UserName, hi welcome to Eastwood High! We like to pair our new students with a friend so that way the transition to a new school is smooth!

Narrator:

Initialize the variables! You think that this idea is:

word idea = "Stupid. Hopefully I don't get a weirdo..."

word idea = "Nice. Now I don't have to worry about making friends!"

(A1.1, A1.3)

Principal:

Also, my office is always open to you.

Seriously, just drop by if you are ever confused or need some food.

I keep my office stocked with candy and sandwiches. You want one?

User:

Wow, okay the principal is actually cool.

I guess if I actually can't make friends I can become friends with her.

Principal:

Okay, so let me introduce you to your new friend!

She'll give you a tour of the school and help you navigate this year.

Oh look, here she comes!

User:

No way! Is that… your best friend all the way back from 5th grade?

Wait, what's her name again?

Narrator:

word bestfriend_name = "_____"

She runs to hug you! Wow, looks like you'll be starting EastWood High with an actual friend!

(A1.1, A1.3)

Best Friend:

Okay, well let me give you a tour!

Let's just visit three places for now so that I can introduce you to some of my friends before class starts.

In what order do you want to proceed with the tour?

Narrator:

Drag the options to complete the variable assignment!

word first_place

word second_place

word third_place

first_place = “ “

second_place = “ “

third_place = “ “

options: gym, cafeteria, courtyard

(A1.1, A1.3, A1.4)

Bestfriend: This is the gym.

I’m a part of the basketball team and we have our first game this Friday.

You should totally come!

We also have assemblies here.

BestFriend: This is the cafeteria.

Oh look there are sign ups for clubs to join.

The counselor’s recommend that everyone join at least one club because it looks

good on

college applications and stuff.

Let’s go look at them!

Narrator:

clubs: math, coding, writing, basketball, spanish, french

A condition causes something to happen only if certain conditions are met.

For example, under the **condition** that I am hungry **then** I will eat.

If I were to write that in a program it might look something like this:

condition i_am_hungry equals true

then eat**()**

I will only eat under the condition that I am hungry.

(T1.6)

If i_am_hungry equals false, then I will not eat.

equals is a special operator that checks to see if two things are equal

(T1.5)

Do you notice the parentheses around eat?

eat() is a function.

Functions are commonly used to break down a problem into smaller pieces.

(T1.9)

More on this later! Let's continue with the story!

Select the appropriate conditional to fulfill your decision:

condition join_math_club equals true

then sign_up_for_math_club**()**

condition join_coding_club equals true

then sign_up_for_coding_club**()**

condition join_spanish_club equals true

then sign_up_for_spanish_club**()**

condition join_basketball_club equals true

then sign_up_for_basketball_club**()**

condition join_writing_club equals true

then sign_up_for_writing_club**()**

(A1.5, A1.6, A1.9)

Narrator:

Great, you're now signed up for clubName!

BestFriend:

You're totally going to make a lot of friends in that club.

Oh, I almost forgot! What lunch are you in?

Perfect! We're in the same lunch section!

BestFriend:

This is the courtyard! Oh look, my friends are hanging out here! Let me introduce you!

This is Jerry and this is Moriah. We've been best friends ever since I moved to Eastwood High in sixth grade.

Moriah:

Hey I'm Moriah! I'm so excited to meet you!

User to herself:

Wow, Moriah is very peppy and happy. She seems like a good friend to have.

Narrator:

How do you want to greet Moriah?

shake_hands(Moriah)

say_hi(Moriah)

hug(Moriah)

What do shake_hands(Moriah), say_hi(Moriah), and hug(Moriah) have in common?

Select your answer below:

They contain parentheses so they must be functions

They are variables

They are conditionals

(A1.9)

Jerry:

Hey I'm Jerry

User to herself:

Jerry is kind of cute. Eastwood High is getting pretty exciting…

Narrator:

How do you want to greet Jerry?

shake_hands(Jerry)

say_hi(Jerry)

hug(Jerry)

User:

It's seriously so nice to meet you guys!

Jerry:

Yeah nice to meet you as well.

Moriah:

Oh yeah we are your best friends now!

Jerry:

What class do you have for first period?

User:

Coding with Mr. Rogers

Moriah:

You're lucky because Mr. Rogers is seriously the best coding teacher at EastWood High.

Jerry:

Wait, no way I have coding with Mr. Rogers for the first period too.

Here I'll walk with you

BestFriend:

Okay bye userName!

I've got French with Mrs. Smith so I'll meet you after the first period!

User: Okay sounds great!

Jerry:

So where are you from?

Narrator:

word hometown = " "

Is **word** a **variable** or **data type**?

variable data type

Is hometown a **variable** or **data type**?

variable data type

(A1.1, A1.2, A1.3)

User:

Well I'm originally from hometown_name but had to move because of my mom's work.

Jerry:

Ah okay, how long did you live there for?

Narrator:

number years =

Is **number** a **variable** or **data type**?

variable data type

Is years a **variable** or **data type**?

variable data type

variable data type

(A1.1, A1.2, A1.3)

Jerry:

Wow, that's a pretty long time. Was it sad for you to leave your friends behind?

User:

Yeah, it was pretty sad. But now I feel so much better now that I've met you guys.

Narrator:

Bell rings

Mr. Rogers:

Alright ladies and gentlemen. Everyone please take a seat.

Jerry:

Looks like I start off my morning with my favorite class everyday

Coding with Mr. Rogers is seriously the best class at EastWood High

User to herself:

Jerry is really nice and seems to be good at school

This will be a great friendship

Jerry:

Look over there, there's two empty seats. Let's grab them!

Narrator:

Jerry turns to you in coding class and bets you that he can convince Mr. Rogers to let everyone out early.

He wants to know what you will bet.

User whispering to Jerry:

There's no way! It's only the first day of school.

Jerry:

How much do you want to bet? Drag the function to complete the condition into the space to complete the statement!

Narrator:

condition (class_leaves_early **equals** true)

then

bet_Jerry(five_dollars)

bet_Jerry(nothing)

bet_Jerry(dessert)

(A1.5, A1.6, A1.9)

Jerry walks up to Mr. Rogers and leaves the classroom.

Looks like Jerry chickened out.

Moments later the fire alarm goes off.

Wait!?!? Did Jerry do this?

Mr. Rogers:

Okay, everyone line up in an orderly fashion, we need to evacuate!

Narrator: There's chaos everywhere…

Everyone starts rushing towards the nearest doors….

User to herself:

Well looks like all of those fire drills we've had are pointless….

Narrator:

Time for lunch…

FINALLY! It's time for lunch and you can finally meet up with bestfriend_name and Moriah to tell them all the crazy things that have happened.

What do you want for lunch? Drag your choice in the underlined area!

type **Drink:** water soda coffee

type **Food:** pizza burrito sandwich

type **Food:** salad pasta soup

type **number:** one two three

Food entree = _____

Drink beverage = _____

number num_fruit = _____

(A1.1, A1.2, A1.3, A1.4)

BestFriend:

UserName! Over here!

User to herself:

Oh great, Moriah and bestfriend_name.

I can finally tell them what has happened

Moriah:

Hey can you believe what happened?

BestFriend:

Yeah, I know! It's so crazy. I wonder who did it!

Moriah:

Seriously whoever did it is a legend!

BestFriend:

Wait, where's Jerry?

Also, why are you so quiet UserName?

User:

Well….

Narrator:

A boolean is a notation used to represent a logical proposition.

In programming, a boolean can be true or false.

Do you tell Moriah and bestfriend_name the truth?

boolean tell_the_truth = false

boolean tell_the_truth = true

(A1.2, A1.2, A1.3)

Narrator:

We can combine a **conditional statement** with an **alternative statement**.

This will cause one thing to happen if a certain condition is met or an alternative thing to happen if it is not met.

For example:

condition it_is_raining equals true

wear_a_raincoat()

alternative

leave_it_at_home()

If it_is_raining, then it will cause wear_a_raincoat() to be executed.

If it is not raining, then it will cause leave_it_at_home() to be executed.

leave_it_at_home() will always be executed if it_is_raining equals false.

(T1.5, T1.7)

Let's continue with the story!

Complete the following to reflect what actions will occur depending on the decision you make

condition(tell_the_truth equals true)

then

alternative

explainWhatHappened()

keepSilent()

(A1.5, A1.7, A1.9)

If user selects **boolean** tell_the_truth equals false

Moriah: Come on, you definitely know something!

bestfriend_name: Yeah, you have to tell us!

User:

Okay fine…

Jerry did it!

Moriah and bestfriend_name:

WHAT!?

User:

Yeah he bet me that he could convince Mr. Rogers to let us out of school early.

BestFriend:

Wow classic Jerry. He's always pulling pranks like these.

But I can't believe that he's already started on the first day of school

Time for the rest of school….

User to herself:

Woah, it's been a long day of school with no sign of Jerry.

Narrator:

A loop will cause something to happen while a certain condition is met.

For example:

loop in_bed equals true

then sleep()

I will sleep for however long I am in my bed.

However, when in_bed becomes false,

I will break out of the loop and sleep() will no longer execute.

(*T1.5, T1.8, T1.9)*

Time to go to homeroom, which you also have with Jerry and Mr. Rogers.

You wonder if Jerry will be there.

Let's continue with the story!

Mr. Rogers:

Class, take a seat. I have an important announcement to make.

User to herself:

There Jerry is, I'm going to go sit by him.

Narrator:

Mr. Rogers has told the class that no one can leave the classroom until the principle finds out who set the fire alarm.

Let's represent the situation with a while loop.

Drag your answer into the space after **loop**!

loop _____

then everyone_stays_in_classroom()

alarm_setter_is_known equals true

alarm_setter_is_known equals false

(A1.3, A1.5, A1.8, A1.9)

Narrator:

You look at Jerry and give him a disapproving look.

Jerry:

Looks like I won the bet. You owe me (whatever bet user selected earlier).

User:

I can’t believe you did that.

It’s only the first day of school and the whole school is already in trouble.

Jerry:

Are you mad?

Jerry:

No one should find out though. Are you planning on telling anyone?

User:

Under one condition

condition

then keep_it_a_secret()

promise_not_to_pull_more_pranks equals true

buy_me_lunch equals true

(A1.5, A1.6)

Jerry:

You got it. Seriously, thank you for keeping my secret.

Narrator:

You look around and everyone is starting to get annoyed and demanding to know what happened…

You have a choice.

You can rat out Jerry and save everyone from having to suffer but then you risk losing Jerry as a friend.

You make eye contact with Jerry.

Do you tell the teacher?

word tell_teacher = “no”

word tell_teacher = “yes”

(A1.3)

Suddenly the principal comes on the intercom….

“Jerry, please come to the office immediately and bring your stuff with you.”...

Wait.. how could he possibly know?….

Jerry doesn’t look too phased. He stands up and gathers his stuff.

Narrator:

A few moments later, Mr. Rogers announces that class is dismissed.

User to herself:

Wow, this has been a crazy day. I can’t wait to go home.

Environment 2

Learning Outcomes: Abstraction and Encapsulation

After completing the second environment the user should be able to :

1. Understand function implementation
2. Understand function parameters
3. Know what abstraction is
4. Know what a class is
5. Know what encapsulation is
6. Know what an object is
7. Understand the dot operator

8. Understand function return types
9. Understand that a parameter list can contain zero or many values
10. Understand variable manipulation

Pretest question #1:

Which of the following contains the function implementation?

word get_name()

word get name()

return my_name

void set_name (word a_name)

none of the above

Pretest question #2:

Which of the following is the function parameter: void draw (**word** object)

void

draw

object

Pretest question #3:

Which of the following is an example of abstraction?

Knowing how to use a computer

Knowing the inner workings of a computer

The procedure of how a computer is built

Pretest question #4:

What is a class?

A data type that can be defined by the programmer

A data type that cannot be defined by the programmer

A variable defined by the programmer

A variable that cannot be defined by the programmer

Pretest question #5:

What is encapsulation?

The separating of an object's data

The bundling of an object's data into a single unit

Allowing an object's data to be accessed by anyone

Pretest question #6:

What is an object?

An instance of a class

Another name for a class

Another name for data type

Pretest question #7:

Why do we use the dot operator?

To access the data of an object

To access only the functions of an object

To access only the functions of an variable

Pretest question #8:

Which of the following is the return type of the following: Person set_name (word name)

Person

set_name

word name

Pretest question #9:

Which of the following is an appropriate number of parameters that a parameter list can have?

0

0 and 1

0, 1 and 2

0, 1, 2…. 20

all of the above

Pretest question #10:

a = 10

a = a + 5

After the second operation is performed, what value will be stored in a?

0

5

10

15

all of the above

Abstraction is a general model of something.

It allows us to use something without knowing the internal workings of it.

Abstraction occurs in many aspects of our daily lives.

Let's consider phones to illustrate this point.

While most of us know *how to use* a phone, most of us *don't know the inner parts* of a phone or *how everything works* within a phone to produce its functionality.

This is a perfect example of abstraction.

(T2.3)

Let's apply our new knowledge!

Which of the following is an example of abstraction?

Knowing how to use a computer

Knowing the internal workings of a computer

(A2.3)

If we want to organize the **data** or information within a phone together, we can represent its **data** within a **class**.

A **class** is a **data type** (like a **word, number,** or **boolean)** defined by the programmer.

Let's look at an example of a class using like the following:

```
class phone {
        boolean touch_screen
        word company
        number year

        void call()
        void ring()
}
```

(T2.4)

Now we know the internal workings of a phone.

Notice that within this **class**, we have **variables** and **functions**.

This bundling of an **object's** data is called **encapsulation.**

Encapsulation is also used to "hide data." More on this later.

I know this is a lot of information, but don't worry! By the end of the episode you'll have a great understanding!

Now, let's get back to the story!

Narrator:

While you're waiting in the courtyard for your mom to pick you up, you decide to look for bestfriend_name and Moriah.

User:

There they are!

Guys, this has been a crazy first day of school!

I'm not sure how they found out that Jerry pulled that alarm.

Moriah:

I know, I honestly cannot believe it! Eastwood High has never been so exciting.

Best_Friend:

Here, let's exchange numbers so we can talk about this when we get home.

Narrator: We are going to represent your best friends within a **class**.

```
class BestFriendGroup {

    word bestfriend_name_phone_number
    word Jerry_phone_number
    word Moriah_phone_number

    void call_bestfriend_name()
    void call_Jerry()
    void call_Moriah()
```

void text_bestfriend_name(**word** message)

void text_Jerry(**word** message)

void text_Moriah(**word** message)

void text_group(**word** message)

void set_bestfriend_name_phone_number(**word** phone_number)

void set_Jerry_phone_number(**word** phone_number)

void set_Moriah_phone_number(**word** phone_number)

}

(T2.4)

Let's set bestfriend_name's number.

To do this, we need to make an **instance** or **object** of the BestFriendGroup **class**.

The **class** is like a blueprint or template of a house- it tells us the basic layout of a house.

The **object** is the actual house- it contains all of its actual properties.

In order to actually have a BestFriendGroup,

we have to create an **instance** of the **class** BestFriendGroup:

BestFriendGroup bff

(T2.4, T2.6)

Weather temperature

Is **Weather** a class or instance?

class instance

Weather temperature

Is temperature a class or instance?

class instance

Student Johnathon

Is **Student** a class or instance?

class instance

Student Johnathon

Is Johnathon a class or instance?

class instance

Drink beverage

Is beverage a class or instance?

class instance

Drink beverage

Is **Drink** a class or instance?

class instance

(A2.4, A2.6)

Notice that **class instantiation** (the creation of an object from a class) is similar to **variable declaration**

Great, now your best friend's are official.

Let's set their numbers.

To access the data of an **object** we use the **dot operator**.

Let me show you an example.

Moriah:

My number is "1923414291"

To set Moriah's number we must use the following **function:**

void set_Moriah_phone_number(**word** number)

(**word** phone_number)

Is **word** a data type or variable?

data type variable

Is phone_number a data type or variable?

data type variable

(A1.1, A1.2)

Perfect!

If you've noticed from the previous examples, whenever we initialize or assign a variable of **data type word**, we surround the **value** with quotations.

(A1.1, A1.2)

Since **variable** phone_number is of **data type word**, when we pass it's value into the function, we have to surround it with quotations as well.

To set Moriah's phone number we do the following:

bff.set_Moriah_phone_number("1923414291")

(T2.7)

Notice that the **data type** is not included when we **pass parameters** in a **function**.

When we **pass parameters** into **functions** we do not include **the data type**.

(T2.2, T2.6)

If we look at the implementation of **class** BestFriendGroup we will find that the highlighted line now contains the **word:** "1923414291" for the variable Moriah_number of **object** bff.

class BestFriendGroup {

word bestfriend_phone_number

word Jerry_phone_number

word Moriah_phone_number

void call_bestfriend()

void call_Jerry()

void call_Moriah()

void text_bestfriend(**word** message)

void text_Jerry(**word** message)

void text_Moriah(**word** message)

void text_group(**word** message)

void set_bestfriend_name_phone_number(**word** phone_number)

void set_Jerry_phone_number(**word** phone_number)

void set_Moriah_phone_number(**word** phone_number)

}

How does this happen? Well **functions** have internal workings.

void set_Moriah_number(word number)

void means that no data is returned. More on that later.

The function implementation is:

void set_Moriah_number(**word** phone_number)

Moriah_phone_number = phone_number

(T2.1)

phone_number is a placeholder that will be assigned to whichever value that you pass in.

When you call bff.set_Moriah_phone_number(“1923414291”) it assigns Moriah_phone_number to the phone_number that you pass in within the parentheses.

Now let’s set bestfriend_name’s number!

Click on the appropriate function in the class details to set bestfriend_name’s number

```
class BestFriendGroup {

    word bestfriend_name_phone_number
    word Jerry_phone_number
    word Moriah_phone_number

    void call_bestfriend_name()
    void call_Jerry()
    void call_Moriah()

    void text_bestfriend_name(word message)
    void text_Jerry(word message)
    void text_Moriah(word message)
    void text_group(word message)

    void set_bestfriend_name_phone_number(word phone_number)
    void set_Jerry_phone_number(word phone_number)
```

void set_Moriah_phone_number(**word** phone_number)

}

Perfect, now drag the appropriate box to set bestfriend_name's number:

bff**.**

set_bestfriend_name_phone_number(**word** "8917264421")

set_bestfriend_name_phone_number("8917264421")

set_bestfriend_name_phone_number(8917264421)

(A2.2, A2.7)

Great! Remember that we do not pass the **data type** when we **pass arguments** into functions!

Moreover, a value of **data type word** must be surrounded by quotations

User:

Wait, what's Jerry's number?

Moriah:

Oh let me check!

It's 3451238971

Narrator:

Set Jerry's number!

Drag the appropriate box after the dot operator.

bff**.**

set_Jerry_number(**word** “3451238971”)

set_Jerry_number(“3451238971”)

set_Jerry_number(**word** 3451238971)

set_Jerry_number(3451238971)

(A2.2, A2.7)

HONK!

Narrator:

You turn around and see your mom.

User:

Bye guys! I’ll text you!

Mom:

Hey user_name, how was the first day of school?

User:

Crazy to say the least…

Narrator:

Text Message from bffs…

Let’s observe the BestFriendGroup class again.

class BestFriendGroup {

word bestfriend_name_phone_number

word Jerry_phone_number

word Moriah_phone_number

void call_bestfriend_name()

void call_Jerry()

void call_Moriah()

void text_bestfriend_name(**word** message)

void text_Jerry(**word** message)

void text_Moriah(**word** message)

void text_group(**word** message)

void set_bestfriend_name_phone_number(**word** phone_number)

void set_Jerry_phone_number(**word** phone_number)

void set_Moriah_phone_number(**word** phone_number)

}

Notice that some function names are preceded by "**void**" and others by "**word**."

The term in front of each **function name** is called a **return type**.

A **return type** can be a **word**, **number**, **boolean**,

or any programmer defined **class** like BestFriendGroup or Phone

a **return type of:**

void means that the function does not return anything.

number means that the function returns a number.

word means that the function returns a word or series of words.

BestFriendGroup means that the function returns an **object** of type BestFriendGroup.

Notice that:

void set_bestfriend_name_number(**word** number)

void set_Jerry_number(**word** number)

void set_Moriah_number(**word** number)

have a **return type** of **void.**

However,

word text_bestfriend_name(**word** message)

word text_Jerry(**word** message)

word text_Moriah(**word** message)

word text_group(**word** message)

have a **return type** of **word**.

This is because whenever you text your friend, you expect them to return a message back to you.

(A2.8)

Now let's get back to texting your best friends!

Jerry:

Hey guys

Select one of the following messages to text:

bff.text_group("Wow, you're alive. I was getting worried you got into some serious trouble")

bff.text_group("Hey")

bff.text_group(Hey)

bff.text_group("Jerry I seriously disapprove of what you did")

bff.text_group(Jerry I seriously disapprove of what you did)

User:

(whatever message selected above)

Moriah:

Jerry OMG

Best friend:

Jerry! Is it true that you pulled the fire alarm?

Jerry:

Oh yeah. I'm a legend.

Best friend:

I seriously cannot believe that you did that.

You are such a troublemaker.

Jerry:

Yeah well the principal and my parents agree.

I've got detention for two months,

and my parents are forcing me to audition for the musical.

They think it'll be good for me.

Select one of the following messages to text:

bff.text_group(**word** "Oh I'm sorry about the detention")

bff.text_group(**word** "Well you deserve it")

bff.text_group("Oh I'm sorry about the detention")

bff.text_group("Well you deserve it")

Jerry:

Yeah well I'm pretty proud of myself.

My record for weeks in detention was

previously 6 weeks but I've finally beat it!

Moriah:

Jerry, that is just not something to be proud of.

Jerry:

Yeah, well we've all got different priorities.

Anyways… anyone want to join the musical with me?

Select one of the following messages to text:

bff.text_group("Uh no way")

bff.text_group("Um yes! I love musicals!")

bff.text_group(**word** “Uh no way”)

bff.text_group(**number** “Um yes! I love musicals!”)

(A2.2, A2.7)

User:

(whatever message selected above)

Bestfriend:

Uhh no way

Moriah:

What, come on bestfriend_name! Musical’s are so much fun!

Jerry:

Yeah, please! If we all do it, it’ll be so much fun!

Bestfriend:

Hmm...Well what musical is it?

Jerry:

Highschool musical

User:

Oh, I love that movie! Okay, I am totally down!

Bestfriend:

Well I guess if everyone is doing it, I’ll do it.

User:

When are auditions?

Jerry:

Friday after school.

Moriah:

I'm so excited! I am finally going to make my acting debut!

Just wait until I'm rich and famous.

Jerry:

Yeah… okay Moriah…

Narrator:

It's Friday….

You're in homeroom….

Mr. Roger:

Okay class, this is the last day of mandatory silence.

Sorry to have to punish everyone like that for the actions of one person.

But for the last day please be silently productive.

Narrator:

Everyone looks at Jerry

Turns out once everyone found out it was Jerry who pulled the fire alarm,

everyone's anger went away.

Everyone ended up thinking that it was funny.

User to herself:

I actually don't mind this mandatory silence.

I've gotten so much homework done.

Let me do my coding homework

Coding homework question #1:

Which of the following is a good example of abstraction?

1. The procedure a surgeon follows when he performs surgery
2. The inner workings of how a phone gets a signal
3. Knowing how to use a microwave

(A2.3)

Coding homework question #2:

What is encapsulation?

1. The bundling of an object's variables and functions
2. Knowing how to use an object but necessarily it's inner workings
3. Having no cohesion of an object's characteristics

(A2.5)

Coding homework question #3:

Do we pass data types as arguments in functions?

1. No
2. Yes
3. Sometimes

(A2.2)

Narrator:

You got (questions correct) / 3 questions correct!

Time for favorite_class…

Congratulations, you got an A+ on your first test!

Later at musical auditions…

User to herself:

I'm not much of an actor but I've always wanted to be in a musical.

Plus I did choir for six years so I'm a pretty decent singer…

Hopefully I get a good role….

After Auditions:

Narrator: Text Message from bffs…

Moriah: How did auditions go? I think I killed it!

Select one of the following messages to text:

bff.text_group(**word** "I was incredible")

bff.text_group("I was incredible")

bff.text_group(**word** "It was okay")

bff.text_group("It was okay")

bff.text_group(I think I did awful)

bff.text_group("I think I did awful")

User:

(whatever message selected above)

Jerry:

I definitely did amazing.

I'm not sure why I never auditioned for a musical before.

Can't wait to tell my parents that I got the part of Troy.

Moriah:

I am totally going to be Gabriella

Best_Friend:

Guys don't be cocky.

I'll be lucky if I got a role as "girl who sits at the table and who is noticed by absolutely no one"

User:

Aw, I'm sure you didn't do that bad

Looks like Mrs. Rodriguez just posted the results!

Moriah:

OMG!

Send a picture!

User:

class Musical_Characters {

word Troy = "Jerry"

word Gabriella = "Shelby"

word Mrs.Darbus = "Moriah"

word girl1 = "bestfriend_name"

word girl2 = "user_name"

word troy_understudy

word Gabriella_understudy

void set_troy_understudy(**word** name)

void set_gabriella_understudy(**word** name)

}

Best_Friend:

Wow, I am seriously only girl2. I don't even have a name.

Jerry:

Hey don't worry about it.

This is only a blueprint.

She hasn't instantiated this class yet, so

so far no one actually has these roles.

(A2.4, A2.6)

Moriah:

Yeah true. user_name and bestfriend_name don't freak out yet.

I, on the other hand, am so excited!

I can't be mad that I didn't get the role of Gabriella because this is my simply my

acting debut.

I should drop out of school and become an actress!

Picture this. Moriah Rhona: The QUEEN of Hollywood

User:

Yeah Moriah… that would be a great idea…

But don't forget you don't actually have the role until this class is instantiated

User to herself: Well I'm kind of disappointed.

But I guess it's whatever.

As long as I get to hang out with my friends it'll be fun.

But who's this Shelby girl?

It seems like Jerry and her are going to get super close.

Not that I'm jealous or anything...

Narrator:

Select one of the options to find out more information about Shelby:

bff.text_bestfriend_name("Hey, just curious… Do you know anything about Shelby?")

bff.call_bestfriend_name()

Bestfriend:

Hey user_name. I guess we're going to be girl1 and girl2 together.

I wish the musical was The Cat and the Hat so we could be Thing1 and Thing 2.

Anyway, Shelby has been doing theater since literally preschool.

She's dominated the musicals every year.

Why did you want to know?

User:

Oh, I'm just excited for her

Narrator: Later in the night….

You have an hour before your bedtime.

Let's express all potential behaviors that you can do as a class called Nighttime.

```
class Nighttime {
        private number homework_grade
        private number art_grade
        private number music_grade
        private number organization_skill

        void do_homework(number time_spent_working)
        void practice_art(number time_spent_working)
        void lisisten_to_music(number time_spent_working)
        void clean_room(number time_spent_working)
}
```

Notice that homework_grade, art_grade, music_greade, and organization_skill contain the keyword **private.**

This means that these variables are only visible to the class that they belong to,

and that these variables cannot be accessed by the dot operator by an object.

For instance, if I had an **instance** of **class** Nighttime called night,

night.homework_grade would result in an error

This is known as "data hiding," a feature of encapsulation.

(T2.5)

Notice that we pass in a **parameter** of **data type number** into each function within the Nighttime **class**.

Each function has a **parameter list** that is enclosed by parenthesis.

A **parameter list** can have zero or many values.

If no values are passed to the function, its parameter list is empty.

(*T2.9)*

Let's look at the **body** of our functions

```
void do_Homework(number time_spent_working)
    number time_passed = 0
    loop (time_passed less than time_spent_working)
        then time_passed = time_passed + 1
        then homework_grade = homework_grade + 1
```

What does all of this mean?

Well first, let me ask you a question.

How long, expressed in minutes, do you want to spend working on homework?

time_spent_working = 5

time_spent_working = 10

time_spent_working = 15

time_spent_working = 20

Okay, so you want to spend *chosen_time* minutes on homework

This means we **call** the function as follows:

do_Homework(*chosen_time*)

(T2.2)

Think of **number** time_passed like a stopwatch that will keep incrementing until it equals chosen_time.

All stop watches stop at time 0 which is why time_passed starts at time 0

Our stop watch is a little different than a typical stopwatch, however.

Instead of increasing by seconds, it increases by minutes.

We do time_passed = time_passed + 1 because we want to increase the stop watch by 1 minute each time we loop

We **loop** while the **condition** time_passed is less than time_spent_working is met

Like equals, less than is a relational operator

Once the condition becomes false, we stop **looping**

(T1.5, T1.8)

Also, for each minute you spend on homework, you're homework grade increases by one point.

That's why we do: homework_grade = homework_grade + 1

The do_homework function has a **return type** of void because it does not need to return anything.

(T2.8, T2.10)

Let's fill out the body of the practice_art function! Enter the highlighted information

```
void practice_art(number time_spent_working)
    number time_passed = [redacted]
    loop (time_passed less than time_spent_working)
        time_passed = time_passed + 1
        art_grade = art_grade + 1
```

Let's fill out the body of the listen_to_music! Enter the highlighted information

```
void listen_to_music(number time_spent_working)
    number time_passed = [redacted]
    loop (time_passed less than time_spent_working)
        time_passed = time_passed + 1
        music_grade = [redacted] + 1
```

Let's fill out the body of the clean_room function! Enter the highlighted information

```
    void clean_room(number time_spent_working)
        number time_passed = [redacted]
        loop (time_passed less than time_spent_working)
            [redacted] = time_passed + 1
            organization_skill = [redacted] + 1
```

(A1.5, A1.8, A2.1, A2.8, A2.10)

Perfect!

You have *60* minutes left.

How else do you want to spend your time? Pick a choice from both columns.

do_homework(30)

practice_art(30)

lisisten_to_music(30)

clean_room(30)

You have *30* minutes left.

How else do you want to spend your time? Pick a choice from both columns.

do_homework(30)

practice_art(30)

lisisten_to_music(30)

clean_room(30)

(A1.5, A1.8, A2.1, A2.8, A2.10)

Narrator:

The next day at school…

It looks like you have a pop quiz in English class!

Good thing you spent time doing your homework yesterday!

You aced it!

In math class….

Jerry:

Did you see that Mrs. Rodriguez posted the list officially by **instantiating** the class Musical_Characters?

class Musical_Characters {

word Troy = “Jerry”

word Gabriella = “Shelby”

word Mrs.Darbus = “Moriah”

word girl1 = “bestfriend_name”

word girl2 = “user_name”

word troy_understudy

word gabriella_understudy

void set_troy_understudy(**word** name)

void set_gabriella_understudy(**word** name)

}

Narrator:

Wait, remind me… How do I instantiate a class again?

Musical_Characters characters

characters **Musical_Characters**

(A2.4, A2.6)

User:

Ah congrats Jerry! That’s so exciting

Jerry:

Thank you! Sorry about your role though.

User:

No, that’s okay. Not everyone can be a star I guess.

I can’t believe I really thought I was going to get a good part.

Jerry:

Hey, Mrs. Rodriguez still hasn’t set the understudy for Gabriella yet.

This is because neither troy_understudy nor gabriella_understudy

have default values.

Who knows, maybe she thinks you are amazing and wants you to be the backup star.

Wait until she sets the understudies to be disappointed.

User:

That's definitely true

Narrator:

Later in the homeroom….

User to herself:

Time to do my coding homework!

Coding homework question #1:

What is a return type?

1. The data type of the value returned from a function
2. The data type of the value returned from a variable

(A2.8)

Coding homework question #2:

Which of the following (if any) is not a valid return type?

1. word
2. void
3. number
4. A programmer defined data type
5. all are valid

(A2.8)

Coding homework question #3:

Look at the following function: **void** goHome (**number** time, **word** friend)

Which of the following is the parameter list

1. void
2. goHome
3. (**number** time, **word** friend)

(A2.9)

Coding homework question #4:

Can private functions be accessed by the dot operator?

1. yes
2. no

(A2.5)

Narrator:

You got (questions correct) / 4 questions correct!

Later at the musical rehearsal…

Mrs. Rodriquez:

Hi everyone, congratulations on landing a role within the musical!

I saw potential and talent within each and one of you!

I want to start rehearsal by announcing the understudies.

The understudy for Troy is Walter Delfino.

Narrator:

Can you set Walter Delfino as the understudy for Troy? Choose the correct option!

characters.set_troy_understudy(**word** “Walter Delfino”)

characters.set_troy_understudy(“Walter Delfino”)

Musical_Characters.set_troy_understudy(“Walter Delfino”)

set_troy_understudy(**word** “Walter Delfino”)

(A2.2, A2.4, A2.6, A2.7, A2.9)

Perfect!

Mrs. Rodriquez:

And the understudy for Gabriella is user_name

Narrator:

Can you set yourself as the understudy for Gabriella? Choose the correct option!

characters.set_gabriella_understudy(**word** “first + last”)

characters.set_gabriella_understudy(“first + last”)

Musical_Characters.set_gabriella_understudy(“first + last”)

set_gabriella_understudy(**word** “first + last”)

(A2.2, A2.4, A2.6, A2.7, A2.9)

Bestfriend:

Oh my goodness, congrats user_name

Moriah:

Wow we are going to Hollywood together!

Jerry:

See, I told you not to worry!

User:

Wow, thanks guys. I am so happy.

I guess my five years of choir really did pay off.

But also, don’t forget I’m only the understudy

Narrator:

Remember that abstraction allows us to use something without knowing the internal workings of it.

Setting the understudies for Troy and Gabriella were perfect examples of abstraction. We were able to use the functions without exactly knowing how they worked.

(T2.3)

Let's look at the function body to understand how they work!

```
class Musical_Characters {
        word Troy = "Jerry"
        word Gabriella = "Shelby"
        word Mrs.Darbus = "Moriah"
        word girl1 = "bestfriend_name"
        word girl2 = "user_name"
        word troy_understudy
        word Gabriella_understudy

        void  set_troy_understudy(word name)
                troy_understudy = name

        void  set_gabriella_understudy(word name)
                [redacted]
}
```

The variable troy_understudy belongs to the **class** Musical_Characters.

It will be assigned the value of whatever value we pass into the **function** as a **parameter**.

Can you fill out the body of **void** set_gabriella_understudy(**word** name)?

(A2.1, A2.2, A2.3)

Narrator:

Looks like it's time to go home!

Your parents said that you may invite one friend over.

You can select the time you leave to go home and the friend that you invite over by using the following function:

void goHome (**number** time, **word** friend)

Like in our other functions, time is expressed in minutes.

If you select time = 0, then you will leave immediately.

If you select time = 10, then you will leave in 10 minutes.

Whatever **value** you **assign** to friend will be invited over to your house.

Drag the following values to complete the function call

goHome (,)

time	friend
0	bestfriend_name
10	Moriah
20	Jerry

(A2.9)

Notice that the function **void** goHome (**number** time, **word** friend) does not belong to a class.

As a result, we do not need to create an instance of a class and access the function by the dot operator.

(T2.7)

Question: Do we need to use the **dot operator** to manipulate a **variable** or a **function** that does not belong to a **class**?

yes no

Later that evening after hanging out….

You have some free time again

Remember that all of the activities you can perform are expressed within a class called Nighttime

```
class Nighttime {
    private number homework_grade
    private number art_grade
    private number music_grade
    private number organization_skill

    void do_homework(number time_spent_working)
```

```
    void practice_art(number time_spent_working)
    void lisisten_to_music(number time_spent_working)
    void clean_room(number time_spent_working)
}
```

As a refresher, here are the details of the function do_homework

```
Homework do_homework(number time_spent_working)
    number time_passed = 0
    loop (time_passed less than time_spent_working)
        time_passed = time_passed + 1
        homework_grade = homework_grade + 1
```

Wait… how does the body of the practice_art function look again?

Enter the highlighted information

```
[redacted] practice_art(number time_spent_working)
    number time_passed = [redacted]
    loop (time_passed less thantime_spent_working)
        time_passed = time_passed + 1
        art_grade = art_grade + 1
```

Let's fill out the body of the listen_to_music function! Enter the highlighted information

```
[redacted] listen_to_music(number time_spent_working)
```

number time_passed = [redacted]

[redacted] (time_passed less thantime_spent_working)

time_passed = time_passed + 1

music_grade = [redacted] + [redacted]

Let's fill out the body of the clean_room function! Enter the highlighted information

[redacted] clean_room(**number** time_spent_working)

number time_passed = [redacted]

[redacted] (time_passed less than time_spent_working)

[redacted] = time_passed + [redacted]

organization_skill = [redacted] + [redacted]

Perfect!

How do you want to spend your time? Pick a choice from both columns.

(Until time left = 0)

time_spent_working = 5	do_homework(time_spent_working)
time_spent_working = 10	practice_art(time_spent_working)
time_spent_working = 15	listen_to_music(time_spent_working)
time_spent_working = 20	clean_room(time_spent_working)

(A1.5, A1.8, A2.1, A2.8, A2.10)

Narrator:

Looks like it's the day of the musical!

Mom:

We can't wait to see you shine tonight!

Dad:

You are going to do great honey. We are going to be your number one fans.

Brother:

Do I have to go?

Dad:

brother_name be supportive….

If you don't go you cannot play video games tonight or tomorrow night

Brother:

Oops… I mean user_name I'm so excited to see you steal the show as girl2

I think that's what I'm going to call you from now on

User:

Rolls her eyes….

Thanks mom and dad. I'll see you guys tonight!

Narrator:

Text Message from bffs…

Moriah:

GUYS! I”M SO EXCITED!

Best Friend:

Me too! We are all going to kill it!

Jerry:

Yeah me too. It’s going to be epic

Select one of the following messages to text:

bff.text_group(“IDK I’m kind of nervous”)

bff.text_group(IDK I’m kind of nervous)

bff.text_group(“It’s going to be so much fun!”)

bff.text_group(It’s going to be so much fun!)

User: (whatever message selected above)

Moriah: WAIT! I just heard news that Shelby got sick!....

Apparently she had some bad shellfish last night

Best Friend: OMG! user_name this means that you are going to play

Gabriella!

Jerry: user_name SEE! I told you that you are a star!

Select one of the following messages to text:

bff.text_group(**word** “Oh my goodness this is amazing”)

bff.text_group(“Oh my goodness this is amazing”)

bff.text_group(“Oh no, I feel so bad for Shelby. But I am also so excited”)

bff.text_group(Oh no, I feel so bad for Shelby. But I am also so excited)

(A2.2, A2.7)

User:

(whatever message selected above)

Moriah:

I'm heading over to the musical now! I'll see you guys there!

Narrator: Later at the performance….

How do you feel?

word feel = "nervous"

word feel = "excited"

Don't worry, you have this!

Mrs. Rodriguez:

user_name, I am so proud of all the effort you have put in!

I know that you are going to do great

Jerry:

Looks like you are the Gabriella to my Troy

User:

Don't be cheesy

Jerry:

Sorry, I couldn't help it.

Honestly, this has been one of my favorite years at this school.

And rehearsing for this musical with you guys has been so much fun…

Narrator:

Time to walk on stage…

Some snippets from the musical….

We're all in this together….

Stick to the status quo….

Bop bop to the top…

Breaking free....

Narrator: The end!

Environment 3

Learning Outcomes: Inheritance

After completing the third environment the user should be able to :

1. Understand the is-a relationship

2. Understand the child class
3. Understand the parent class

Pretest question #1:

Which of the following is not an appropriate is-a relationship

A car is-a vehicle

A tire is-a vehicle

A baby is-a human

All of the above contain an appropriate is-a relationship

Pretest question #2:

If class A is-a class B then:

class A inherits the functionality of class B

class B inherits the functionality of class A

Neither class inherits the functionality of the other

Pretest question #3:

Does the parent class have access to the functionality of the child class?

Yes

No

Narrator:

The Musical was a great way to end the semester!

Your parents and even your brother were very proud of you!

Brother:

Wow username you were actually so good! I can't believe I wanted to miss this!

Dad:

Seriously, my little star!

Here you go, we got you some flowers!

user:

Aww thank you guys!

Mom:

The best Gabriella I've ever seen!

You know, you and brother_name have been adjusting very well to Eastwood High…

You both have stellar grades and you have become so involved….

Your dad and I were talking and we think it's time we finally get you guys a pet

Brother:

WHAT!?! No way!!

Can we get a lizard? Or maybe a pig? Or maybe a corn snake I've always wanted one of those

user:

Or maybe even a dog or a bird?

Brother:

Ooh yeah, I've always wanted a dog

Dad:

Okay guys we're going to have to take you to the pet store so you can decide then

Narrator: Later at the pet store…

Wow, look at all of these pets!

There are hamsters, dogs, cats, chinchillas, snakes,

birds and lizards!

You've got so many options to choose from!

Brother:

Aw there aren't any pigs!

Narrator:

In everyday life, you can find many objects that are specialized forms of other objects.

When one object is a specialized version of another object, there is an **is-a** relationship between them.

For instance, a hamster or a dog **is-a** specialized form of a pet.

Because a hamster or a dog **is-a** pet, they have all of the general characteristics of a pet.

In addition, they also have special characteristics of their own.

For example, a dog can bark and a hamster can run on its wheel.

(T3.1)

The **is-a** relationship involves a **general class** and a **derived class**. Let's take a look:

class Pet{

word name

```
        void set_name(word name)
        void play()
        void eat()
        void sleep()
        void drink()
        void use_bathroom()
}

class Dog is-a Pet{
        void bark()
        void chase_cat()
}
class Hamster is-a Pet{
        void use_wheel()
        void squeal()
}
```

class Pet is the **general class** and **class** Dog and **class** Hamster are the **derived classes.** Because a Dog **is-a** Pet because a Hamster **is-a** Pet, an object of type Dog and Hamster can play(), eat(), sleep(), and drink().

However, an object of type Dog can also bark() and chase_cat() while an object of type Hamster can also use_wheel() and squeal();

(T3.2, T3.3)

Let's visit some of the animals.

You can select the animal you visit next via the **function** visit_next.

Function visit_next takes in a **number** as a **parameter** and **returns** a **word**.

Let's look at its implementation:

```
word visit_next (number choice)
    if choice equals 1
        return "dog"
    if choice equals 2
        return "cat"
    if choice equals 3
        return "bird"
    if choice equals 4
        return "hamster"
```

What is your choice?

number choice = 1 **number** choice = 2 **number** choice = "1"

number choice = 3 **number** choice = 4 **number** choice = "2"

number choice = "4" **number** choice = "3"

(A1.6 A2.1, A2.2, A2.8, A2.9)

Great! Remember that when a **function returns** a value, we can create a variable to store the **returned** value

(A1.3)

Let's create a variable of the appropriate type to store the value returned from the function

word animal = visit_next (choice)

number animal = visit_next (choice)

(A1.2, A1.3)

(if animal equals "cat")

class Cat **is-a** pet{

    meow()

    run_from_dog()

    purr()

}

Narrator: Okay, let's create an instance of **class** Cat so that you visit one!

    Cat persian_cat

Pet store worker: This is our persian cat. It's 6 months old.

brother_name: Aww it's sleeping.

Narrator: How do we show that the persian cat is sleeping?

Remember that the **general class** Pet looks like this:

```
class Pet{
        word name

        void set_name(word name)
        void play()
        void eat()
        void sleep()
        void drink()
        void use_bathroom()
}
```

and the **derived class** Cat looks like this:

```
class Cat is-a Pet{
        void meow()
        void run_from_dog()
        void purr()
}
```

Select the appropriate answer

persian_cat.sleep() Cat.sleep()

persian_cat.nap() Cat.nap()

persian_cat.sleeping() Cat.sleeping()

(A2.4, A2.6, A3.1, A3.2, A3.3)

Narrator:

Perfect! Even though the **function** sleep() is in the **class** Pet,

because a Hamster **is-a** Pet,

an object of **class** Hamster has access to Pet's sleep() **function**

Let's create another instance of type Cat so we can view another cat.

Cat bengal_cat

Pet Store Worker:

This is our bengal cat.

Dad:

The little guy is adorable. Look, it's purring!

Narrator:

How do we represent that the bengal cat is purring?

Select the appropriate answer

bengal_cat.purr() Cat.purr()

bengal_cat.purring() Cat.purring()

Narrator:

Perfect! The **function** purr() belongs to the **class** Cat,

so any **object** of **class Cat** has access to it.

(A2.4, A2.6, A3.1, A3.2, A3.3)

Mom:

These cats sure look cute! Do you guys want a cat?

Brother:

No I want to keep looking!

You can select the animal you visit next via the **function** visit_next.

Function visit_next takes in a **number** as a **parameter** and **returns** a **word**.

Let's look at its implementation again:

```
word visit_next (number choice)
    if choice equals 1
        return "dog"
    if choice equals 2
        return "cat"
    if choice equals 3
        return "bird"
    if choice equals 4
        return "hamster"
```

What is your choice? (minus the choice previously selected)

number choice = 1 **number** choice = 2 **number** choice = “1”

number choice = 3 **number** choice = 4 **number** choice = “2”

number choice = “4” **number** choice = “3”

(A1.6, A2.1, A2.2, A2.8, A2.9)

(if animal equals “hamster”)

class Hamster **is-a** Pet{

use_wheel()

squeal()

}

Narrator:

Okay, let’s create an instance of **class** Hamster so that you visit one!

Hamster golden_hamster

(A2.4, A2.6)

Pet store worker:

This is our golden hamster. It was born two weeks ago!

Mom:

Wow look at it play with its toy!

It sure has a lot of energy!

Narrator:

How do we show that the hamster is playing?

Remember that the **general class** Pet looks like this:

```
class Pet{

        word name


        void set_name(word name)

        void play()

        void eat()

        void sleep()

        void drink()

        void use_bathroom()

}
```

and the **derived class** Hamster looks like this:

```
class Hamster is-a Pet{

        void use_wheel()

        void squeal()

}
```

Select the appropriate answer

golden_hamster.play()	Hamster.play()
golden_hamster.playing()	Hamster.playing()
golden_hamster.run_around()	Hamster.run_around()

(A2.4, A2.6, A3.1, A3.2, A3.3)

Narrator:

Perfect! Even though the **function** play() is in the **class** Pet,

because a Hamster **is-a** Pet, an object of **class** Hamster has access to Pet's play()

function.

Let's create another instance of **class** Hamster so we can view another hamster.

Hamster dwarf_hamster

(A2.4, A2.6)

Pet store worker:

This is our dwarf hamster.

Look how fast it can run on its wheel.

Dad:

Wow that's fast! I think it should join the Olympics!

User:

Hahaha….

Narrator:

How do we show that the dwarf hamster is running on its wheel?

Remember that the **general class** Pet looks like this:

class Pet{

word name

void set_name(**word** name)

void play()

void eat()

void sleep()

void drink()

void use_bathroom()

}

and the **derived class** Hamster looks like this:

class Hamster **is-a** Pet{

void use_wheel()

void squeal()

}

Select the appropriate answer

dwarf_hamster.run_on_wheel()	Hamster.run_on_wheel()
dwarf_hamster.use_wheel()	Hamster.use_wheel()

Narrator: Perfect! the **function** use_wheel() belongs to the **class** Hamster so any **object** of **class** **Hamster** has access to it.

(A2.4, A2.6, A3.1, A3.2, A3.3)

You can select the animal you visit next via the **function** visit_next.

Function visit_next takes in a **number** as a **parameter** and **returns** a **word**.

Let's look at its implementation again:

```
word visit_next (number choice)
    if choice equals 1
        return "dog"
    if choice equals 2
        return "cat"
    if choice equals 3
        return "bird"
    if choice equals 4
        return "hamster"
```

What is your choice? (minus the choices previously selected)

number choice = 1 **number** choice = 2 **number** choice = "1"

number choice = 3 **number** choice = 4 **number** choice = "2"

number choice = "4" **number** choice = "3"

(A1.6, A2.1, A2.2, A2.8, A2.9)

(if animal equals "dog")

```
class Dog is-a Pet{
    bark()
```

chase_cat()

}

Narrator:

Okay, let's create an instance of **class** Dog so that you visit one!

Dog husky_dog

Pet store worker:

This is our husky. It is very rambunctious

Husky to user:

Roof roof!

Brother:

Oh, he's a little loud

Narrator:

How do we show that the husky is barking?

Remember that the **general class** Pet looks like this:

class Pet{

word name

set_name(**word** name)

void play()

void eat()

void sleep()

void drink()

void use_bathroom()

}

and the **derived class** Dog looks like this:

class Dog **is-a** Pet{

void bark()

void chase_cat()

}

Select the appropriate answer

husky_dog.run_on_wheel() Dog.run_on_wheel()

husky_dog.use_wheel() Dog.use_wheel()

(A2.4, A2.6, A3.1, A3.2, A3.3)

Narrator:

Perfect!

The **function** bark() belongs to the **class** Dog so any **object** of **class** Dog has access to it.

Let's create another instance of **class** Dog so we can view another dog.

Dog golden_retriever

Pet store worker:

This is our golden retriever.

It was just running around with the other pups from its litter so it is very thirsty.

Narrator:

How do we show that the golden_retriever is drinking water?

Remember that the **general class** Pet looks like this:

```
class Pet{

        word name


        void set_name(word name)

        void play()

        void eat()

        void sleep()

        void drink()

        void use_bathroom()

}
```

and the **derived class** Dog looks like this:

```
class Dog is-a Pet{

        void bark()

        void chase_cat()

}
```

Select the appropriate answer

golden_retriever.drink() Dog.drink()

golden_retriever.drink_water() Dog.drink_water()

Narrator:

Perfect!

Even though the **function** drink() is in the **class** Pet, because a Dog **is-a** Pet,

an object of **class** Dog has access to Pet's drink() **function**

Let's look at the last pet!

```
word visit_next (number choice)
    if choice equals 1
        return "dog"
    if choice equals 2
        return "cat"
    if choice equals 3
        return "bird"
    if choice equals 4
        return "hamster"
```

What is your choice? (minus the choices previously selected)

number choice = 1 **number** choice = 2 **number** choice = "1"

number choice = 3 **number** choice = 4 **number** choice = "2"

number choice = "4" **number** choice = "3"

(A1.6 A2.1, A2.2, A2.8, A2.9)

(if animal equals “bird”)

class Bird **is-a** Pet{

imitate()

fly()

}

Narrator: Okay, let’s create an instance of **class** Bird so that you visit one!

Bird parrot

Pet store worker: Ooh this is the parrot. It loves to repeat whatever you say.

Brother: (Turns to parrot) I am stupid

Parrot: I am stupid

Mom: Hey that is not very polite

User: brother_name is annoying

Parrot: brother_name is annoying

Dad: Guys...

Narrator: How do we show that the parrot is imitating sounds?

Remember that the **general class** Pet looks like this:

class Pet{

word name

void set_name(**word** name)

void play()

void eat()

void sleep()

void drink()

void use_bathroom()

}

and the **derived class** Bird looks like this:

class Bird **is-a** Pet{

void imitate()

void fly()

}

Select the appropriate answer

parrot.speak()	Bird.imitate()
parrot.imitate()	Bird.speak()

Perfect!

The **function** speak() belongs to the **class** Bird,

so any **object** of **class** Bird has access to it.

(A2.4, A2.6 A3.1, A3.2, A3.3)

Let's create another instance of **class** Bird so we can view another bird.

Bird finch

Pet store worker:

This is our finch.

We need to get its wings clipped because it keeps flying everywhere.

Narrator:

How do we show that the finch is flying?

Remember that the **general class** Pet looks like this:

```
class Pet{
        word name


        set_name(word name)
        void play()
        void eat()
        void sleep()
        void drink()
        void use_bathroom()
}
```

and the **derived class** Bird looks like this:

```
class Bird is-a Pet{
        void imitate()
        void fly()
}
```

Select the appropriate answer

parrot.fly() Bird.fly()

parrot.flying() Bird.flying()

Perfect!

The **function** fly() belongs to the **class** Bird so any **object** of **class** Bird has access to it.

Dad:

Okay so we've seen all of the animals.

Which one do you guys want?

Brother:

I really want a bird!

Narrator:

Do you want a bird? Select your answer

boolean want_a_bird = true **boolean** want_a_bird = false

(if want_a_bird equals true)

Narrator:

Okay, which bird do you want?

word bird = "finch"

word word = "parrot"

(if want_a_bird = false)

Narrator:

Okay, then which animal do you want?

word animal = “dog”

word animal = “cat”

word animal = “hamster”

(if animal equals “dog”)

Narrator: Okay, which dog do you want?

word dog = “golden retriever”

word dog = “poodle”

(if animal equals “cat”)

Narrator: Okay, which cat do you want?

word cat = “persian”

word cat = “bengal”

(if animal equals “hamster”)

Narrator: Okay, which cat do you want?

word hamster = “dwarf”

word hamster = “golden”

Narrator:

Congratulations!

You just adopted a (whichever animal was selected)

Let's set its name.

Enter its name in the quotations in the parenthesis

animal.setName (“ ”)

Mom:

Alright guys, let's go take our new pet home.

Environment 4

Learning Outcomes: Polymorphism

After completing the fourth environment the user should be able to :

1. Understand what polymorphism is
2. Understand the relationship between inheritance and polymorphism
3. Know what override means

Pretest question #1:

What is polymorphism?

The ability to process objects differently depending on their class

The ability for a variable to be assigned several different values

The ability for a function to be called more than once

Pretest question #2:

Can polymorphism exist between class A and class B if they do not have an is-a relationship?

no

yes

yes if we override the classes

Pretest question #3:

What does override mean?

1. It allows the derived class to provide a specific implementation of a method that is provided by parent class

2. It allows the general class to provide a specific implementation of a method that is provided by the parent class

3. It allows class A to access the data of class B if there is not an is-a relationship

Narrator:

You've spent the whole weekend playing with pet_name, but now it's time to go back to school…

Usually your mom drives you and your brother to school, but she has given you the choice to also walk, ride your bike, or take the bus.

A bus **is-a** mode of transportation.

A walk **is-a** mode of transportation.

A bike **is-a** mode of transportation.

A car **is-a** mode of transportation.

Let's take a look at the classes

```
class Mode_of_Transportation{
        word move()
                return “I am moving”
        word brake()
                return “I am stopping”
}

class Bus is-a Mode_of_Transportation{
        word move()
                return “I am moving at 40 mph”
}

class Walk is-a Mode_of_Transportation{
        word move()
                return “I am moving at 4 mph”
}

class Bike is-a Mode_of_Transportation{
        word move()
                return “I am moving at 10 mph”
}
```

class Car **is-a** Mode_of_Transportation{

word move()

return “I am moving at 45 mph”

}

We have an **is-a** relationship like in the pet store scenario.

Notice that the **function name** and **parameter list** in the **derived** classes are the same as the **function name** and **parameter list** in the **general** class.

A **derived** class can **override** a **function** of its **general** class by defining a **function** with the same **function name** and **parameter list.**

(T3.2, T3.3, T4.3)

Let’s use an example of animals to reinforce this concept

First, help me make the classes.

Does an **is-a** relationship between a **class** Dog and class **Animal** make sense (i.e. is a dog an animal)? Select your answer below:

boolean makes_sense = false **boolean** makes_sense = false

Does an **is-a** relationship between a **class** Cat and class **Animal** make sense (i.e. is a cat an animal)? Select your answer below:

boolean makes_sense = false **boolean** makes_sense = false

Does an **is-a** relationship between a **class** Cow and class **Animal** make sense? Select your answer below:

boolean makes_sense = false **boolean** makes_sense = false

Does an **is-a** relationship between a **class** Leash and class **Animal** make sense? Select your answer below:

boolean makes_sense = false **boolean** makes_sense = false

Perfect! A leash is not an animal and so it should not **inherit the** functionality of **class** Animal!

(T3.1)

Now let's look at the actual implementation of the classes below!

```
class Animal {
word make_noise()
        return "I make a noise"

word sleep()
        return "zzzzzzzzz"
}

class Dog is-a Animal{
```

```
        word make_noise()
            return “roof”
}

class Cat is-a Animal{
        word make_noise()
            return “meow”
}

class Cow is-a Animal{
        word make_noise()
            return “moo”
}


class Turtle is-a Animal{
        void make_noise()
}
```

We know that a Dog **is-a** Animal, that a Cat **is-a** Animal, and that a Cow **is-a** Animal. However, dogs, cats, and cows make different noises, so it is necessary that they **implement** their make_noise() function differently.

A dog must bark, a cat must meow, and a cow must moo.

Notice that we do not override the **function** with name and parameter list: sleep()
This is because in our case, all animals make the noise "zzzzzzzzz" when they sleep

Let's practice! Say I have an **instance** of **class** Cow called cow, an **instance** of **class** Cat called cat, an **instance** of **class** Dog called dog, and an **instance** of **class** Animal called animal
If I want to make the object cow "moo," how do I get it to do so? Select your answer from the following:

cow.moo()	animal.moo()
cow.make_noise()	animal.make_noise()

If I want to make the object dog "bark," how do I get it to do so? Select your answer from the following:

dog.make_noise()	animal.make_noise()
dog.bark()	animal.bark()

If I want to make the object dog sleep how do I get it to do so? Select your answer from the following:

dog.sleep()	animal.sleep()
dog.zzzzz()	animal.zzzzz()

If I call **word** sound_made = animal.make_noise() what value will be stored in sound_make? Select your answer from the following:

sound_made equals “I make a noise”

sound_made equals “bark”

sound_made equals “meow”

sound_made equals “moo”

(T3.1, T3.2, T3.3, T4.1, A4.2)

Perfect! Now let’s get back to the story!

Oh no! You’re almost late to school!

Which mode of transportation is the best choice so that you are not late to school?

Let’s revisit the **implementation** again:

```
class Mode_of_Transportation{
        void move()
                return “I am moving”
        word brake()
                return “I am stopping”
}

class Bus is-a Mode_of_Transportation{
        word move()
                return “I am moving at 40 mph”
```

```
}

class Walk is-a Mode_of_Transportation{
        word move()
                return "I am moving at 4 mph"
}

class Bike is-a Mode_of_Transportation{
        word move()
                return "I am moving at 10 mph"
}

class Car is-a Mode_of_Transportation{
        word move()
                return "I am moving at 45 mph"
}
```

Pick the **class** that will get you to school the fastest!

class Mode_of_Transportation **class** Bus

class Walk **class** Bike

class Car

Perfect! Now let's make an instance of **class** Car called car

Wait, do we need to **override** the brake() function in **class** Car? Select your answer from the following:

word answer = “Yes, become some cars do not brake”

word answer = “No, because all cars brake”

(A4.1)

Perfect!

Mom: Alright honey, let’s go! I don’t want you to be late to school.

User: Brother_name hurry up!

Brother: I’m coming!

Narrator: How do we get the object car to move?

class Car.move() car.move()

(A2.4, A3.6)

Perfect! Let’s head to school!

Oh… did I mention that it's Valentine’s Day?

Before first period with Mr.Rogers, you meet with bestfriend_name, Moriah, and Jerry in the the courtyard

Best Friend:

Happy Valentines day guys!

Moriah:

Happy Valentines!

Look! I got a rose delivered to me this morning in the cafeteria…

I got it from Johnathon and we're going to dinner tonight!

bestfriend_name:

Yay I'm happy for you!

I got one too from Sammy!

User:

Aww that's so sweet!

I don't think I'm going to get a rose….

Best Friend:

Hey, I bet you will

But if not we can all do something fun tonight

Moriah:

Yeah, like bake cookies and watch a horror movie

bestfriend_name:

Oh no not horror movies, I hate those!

Moriah:

Ugh fine

User:

Haha thanks guys, that sounds like fun

Jerry:

Hey, I have a pretty good feeling you're going to get a rose today….

Ready to go to coding class?

User:

Yup, see you later guys

Narrator:

Later in coding class….

Mr.Rogers:

Hey everyone! Happy Valentine's Day!

I have two surprises for you…

The first is that I got you all candy

The second is that I'm giving you guys a surprise quiz!

You can take however much candy you want once you turn the quiz in

Jerry to user:

Oh goodness, a pop quiz on Valentine's Day. That's disrespectful

Coding quiz:

Observe the class implementations below:

```
class Shape{
    void identity()
        print_to_screen (“I am a Shape object”)

    void sides()
        print_to_screen (“I have sides”)

}
```

class Rectangle **is-a** Shape{

    void identity()

        print_to_screen (“I am a Rectangle object”)

}

class Glass{

    void identity()

        print_to_screen (“I am a Glass object”)

}

Notice that **class** Glass does not have an **is-a** relationship with **class** Shape

Coding quiz question #1:

If I **instantiate** an **object** of **class** Shape called shape, what is the appropriate output of shape.idenity()?

1. I am a Shape object
2. I have sides
3. I am a Rectangle object

Coding quiz question #2:

If I **instantiate** an **object** of **class** Rectangle called rectangle, what is the appropriate output of rectangle.sides()?

1. I am a Shape object
2. I have sides

3. I am a Rectangle object

Coding quiz question #3:

If I **instantiate** an **object** of **class** Rectangle called rectangle, what is the appropriate output of rectangle.idenity()?

1. I am a Shape object
2. I have sides
3. I am a Rectangle object

Coding quiz question #4:

If I **instantiate** an **object** of **class** Glass called glass, what is the appropriate output of glass.idenity()?

1. I am a Shape object
2. I have sides
3. I am a Glass object

Perfect!

It is important to note that because glass does not have an **is-a** relationship with Shape, it is not **overriding** any of its functions

(T4.3)

Coding quiz question #5:

If I **instantiate** an **object** of **class** Glass called glass, what is the appropriate output of glass.sides()?

1. I have sides
2. I am a Glass object
3. This is an error, a Glass object does not have an **is-a** relationship with a Shape object. Therefore it does not have access to its data.

(A3.1, A3.2, A3.3, A4.1, A4.2, A4.3)

Coding quiz question #6

What does overriding allow us to do?

1. It allows the derived class to provide a specific implementation of a method that is provided by the general class
2. It allows the general class to provide a specific implementation of a method that is provided by the derived class
3. It allows class A to access the data of class B if there is not an is-a relationship

(A4.3)

Great job! You got questions_correct/6 questions correct!

Later in English class….

Rose delivery person:

Knocks on the door door….

Rose delivery! I have a delivery for James… Rosa… Octavis… Julie… and user_name….

User:

Oh wow, who got this rose for me? This is so sweet!

User reads rose:

I am happy you moved to Eastwood High, you have become one of my closest friends. Starring in Highschool Musical with you has been one of the highlights of my year.

Sincerely, Troy

User to herself:

Wait… did Jerry get this for me?

Narrator:

Later after English class….

Bestfriend:

Hey did you get your rose yet?

User:

Yeah I did… Did you know about this?

Best Friend:

Yeah Jerry made me promise not to tell you.

He's been planning this since the musical!

User:

And you didn't tell me? Wow you're a good liar.

Best Friend:

I know! So how do you feel about it?

User: I'm really happy!

Homeroom got cancelled for the assembly though, and since he's performing in the band, I'm not sure when I'm going to see him

Best Friend:

Well I'm pretty sure he has something planned

User:

Like what?

Best Friend:

Sorry… I promised

Narrator:

Later at home….

Dad:

user_name, don't forget to feed pet_name again

User:

Okay, I'm on it

User to pet:

You are so cute, I'm so glad we adopted you

Narrator:

Looks like you just got a text from Jerry!

Remember that you can text Jerry by **instantiating** an **object** of **class** BestFriendGroup like this:

BestFriendGroup bff

Let's look at the implementation of **class** BestFriendGroup again. Which function do you have to call inorder to text Jerry?

class BestFriendGroup {

word bestfriend_name_phone_number

word Jerry_phone_number

word Moriah_phone_number

void call_bestfriend_name()

void call_Jerry()

void call_Moriah()

void text_bestfriend_name(**word** message)

void text_Jerry(**word** message)

void text_Moriah(**word** message)

void text_group(**word** message)

void set_bestfriend_name_phone_number(**word** phone_number)

void set_Jerry_phone_number(**word** phone_number)

void set_Moriah_phone_number(**word** phone_number)

}

(T2.4)

Perfect! Let's get back to the story!

Jerry:

Hey user_name! I'm just wondering if you got my rose?

User:

Hey Jerry! I did and I love it!

Jerry:

(Likes your message) Haha thank goodness

So… I was wondering if you want to go to a restaurant tonight?

User:

Yeah, that would be fun!

Jerry:

Okay great! Do you have a preference for the place?

Narrator: Let's represent the restaurants you can choose from as classes:

```
class Restaurant{
    void order()
        tell_waiter ("I would like to order")
```

```
        void pay(number amount)
                tell_waiter (“I am paying ” + amount)

        word eat(word item)
                return “I am eating” + item
}

class                   is-a Restaurant{
        void order()
                tell_waiter (“I am ordering a          ”)
}

class Coffee_Shop is-a Restaurant{
        void order()
                tell_waiter (“I am ordering a coffee”)
}

class Pizza_Palace is-a Restaurant{
        void order()
                tell_waiter (“I am ordering pizza”)
}
```

class Burrito_Mania **is-a** Restaurant{

void order()

tell_waiter (“I am ordering a burrito”)

}

(T2.4, T3.1, T3.2, T3.3, T4.1, T4.2)

Select one of the following messages to text:

bff.text_Jerry(“Yeah, I’d love to go to a coffee shop!”)

bff.text_Jerry(“Yup, how about pizza palace?”)

bff.text_Jerry(“Hmmm… I’m thinking Burrito Mania!”)

bff.text_Jerry(Yeah, I’d love to go to a coffee shop!)

bff.text_Jerry(Yup, how about pizza palace?)

bff.text_Jerry(Hmmm… I’m thinking Burrito Mania!)

User:

(whichever option selected)

Jerry:

That sounds great! My parents can drive us at 6:30

User:

Sounds great!

(A2.2, A2.7)

Dad:

Aw, you’re all grown up

Mom:

Keep me updated and make sure to text me every ten minutes

User:

Mom… that seems a little unnecessary….

Mom:

Okay fine. But keep me updated!

This is so exciting, let me take another picture

(Pretending the user picked Pizza Palace, the other scenarios will be very similar)

Narrator: Later at Pizza Palace….

Let's **instantiate** an **object** of Pizza_Palace and call it pizza_palace

Narrator:

The conversation is going really well….

And you and Jerry are having a blast….

Let's order some food!

pizza_palace.order()

Is the line of code that begins with "tell_waiter" a function or a variable?

word answer = "function" **word** answer = "variable"

(A1.1, A1.9, A2.2)

Once we call object_name.order(), which **parameter** is being **passed** into the **function**

tell_waiter?

"I would like to order"

"I would like to order pizza"

"I would like to order a coffee"

"I would like to order a burrito"

(A2.1, A4.1)

Perfect!

When pizza_palace.order() is called, is any value returned?

boolean value_returned equals true **boolean** value_returned equals false

(A2.8)

Great!

The food has arrived!

When we call order.eat("pizza") is the eat **function** being called from **class** Pizza_Palace or from **class** Restaurant?

word answer = "Pizza_Palace" **word** answer = "Restaurant"

(A4.1)

Perfect, the eat **function** belongs to the class Restaurant and is not overridden by **class** Pizza_Palace.

Because Pizza_Palace **is-a** Restaurant, Pizza_Palace has access to the data within the class Restaurant.

Time to pay….

Waiter: That will be $12

Narrator: Select the appropriate function

pizza_palace.pay(12) pizza_palace.pay(“12”)

(A1.2)

Perfect!

Narrator:

Valentine’s Day was a success…

You and Jerry both had a wonderful time!

Looks like moving to a new school wasn’t you bad after all…

Who would’ve known that you would’ve rekindled your friendship with bestfriend_name and grew so close to Moriah and Jerry?

And now, it looks like you and Jerry are dating!

What an exciting adventure we’ve had over these past four episodes!

I hope you’ve learned a lot and that the rest of your time at EastWood High goes smoothly!

Looks like my job is over, so it's time for you to continue the story on your own!

Supplemental Module

Five questions will be selected from a bank of ten questions and given to the user for completion. The outcome of the module depends on the user's performance on the quiz with the most optimal outcome resulting if the user answers 5/5 correctly, the moderate outcome resulting if the user answers 3/5 or 4/5 questions correctly, and the least favorable outcome resulting if the user answers 1/5 or 2/5 corrections correctly.

It's finally time for Spring Dance! You and Jeremy have been dating for two months and you both can't wait to go to your first dance together! It's going to be such an iconic night! In order to have the most fun night, try to answer the following 5 questions as accurately as possible!

Questions:

1. You, Moriah, and bestfriend_name have decided to match my wearing red dresses! What is the difference between dress_color = "red" and x equals "red"?

a. dress_color = 7 checks to see if dress_color and 7 are the same value while dress_color equals 7 assigns to dress_color the value of 7

b. dress_color equals 7 checks to see if dress_color and 7 are the same value while dress_color = 7 assigns to dress_color the value of 7

c. There is no difference

2. Your mom told you that if you clean your room the day before the dance, then she will take you to get your nails done. Pick the answer that causes the function nail_salon() to be called if you cleaned your room. If your room is not clean, then do_own_nails() will be called.

a. **loop** (room_is_clean equals true)

then nail_salon

alternative

then do_own_nails()

b. **condition (**nail_salon equals true)

then room_is_clean()

alternative

then do_own_nails()

c. **condition** (room_is_clean equals true)

then nail_salon()

alternative

then do_own_nails()

3. It's time to get ready! Pick the answer that appropriately creates an object of class SpringDance and calls it's getReady() function

    a. SpringDance dance

    dance.getReady

    b. SpringDance dance

    dance.getReady()

    c. SpringDance.getReady()

4. Jeremy has just arrived at your house. Your parents want to take photos for a duration of ten minutes. Select the answer appropriate to demonstrate this.

    a. time = 10

    loop (time not equals 10)

    then take_photos()

    time++

    b. time = 0

    loop (time not equals 10)

    then take_photos()

    time++

    c. time = 0

    loop (time equals 10)

    then take_photos()

    time++

5. After pictures, your parents decided to take you and Jeremy out to eat! For dessert you decided to get pie. Observe the following classes:

```
class Food {
        word name
        number amount
        void eat()
                display ("I am eating")
}

class Pie{
        word name
        number amount
}
```

If I create an instance of class Pie called pie, what happens if I do: pie.eat()

a. "I am eating" will be displayed
b. This is an error because an is-a relationship was not declared between class Pie and class Food

6. Time to go shopping for shoes! If **class** Heels **is-a class** Shoe, then:

a. an object of **class** Shoe can access the functions and variables of **class** Heels
b. an object of **class** Heels can access the functions and variables of **class** Shoe
c. neither is correct

7. is **class** Heels the derived class or general class?

a. derived class

b. general class

8. Moriah and best_friend name ask you if you want to have a sleep over after the dance and of course you do! Select the answer that makes sense.

a. **boolean** sleepover = “yes”

b. **string** sleepover = “yes”

c. **boolean** sleepover equals “yes”

d. **string** sleepover equals “yes”

9. You know how to do your spring dance makeup despite not knowing the actual way the makeup is made. This is an example of:

a. abstraction

b. parametrization

c. overriding

10. Your parents got you a gift but they are keeping it a surprise and won’t tell you. This hiding of data by making it private is an example of:

a. encapsulation

b. abstraction

c. implementation

Best Outcome:

One week before the spring dance:

Principle on the intercom:

Hello everyone, I'd like to announce those in the running for the titles of spring queen and spring king.

For spring queen: Julia Fey, Ellen Kimmel, and user_name.

For spring king: Jimmy Degenerees, Conan Fallon, and Jeremy Moore.

Narrator:

No way! You can't believe that you and Jeremy were both nominated! If you win then you will seriously be the Troy and Gabriella of Eastwood High!

Day of the dance: You and your friends all look stunning! Since you guys arrived at the dance with your dates you've been dancing and eating. This is the best night you've ever had at Eastwood High!

Looks like they're announcing the winners of the title spring queen and spring king!

Congratulations! Both you and Jeremy won the titles! This is such a perfect way to end the semester!

The rest of the night was a blast! Let's look at some pictures!

There are more questions to answer! Play again for more practice!

Moderate Outcome:

One week before the spring dance:

Principle on the intercom:

Hello everyone, I'd like to announce those in the running for the titles of spring queen and spring king.

For spring queen: Julia Fey, Ellen Kimmel, and Tina O'brien

For spring king: Jimmy Degenerees, Conan Fallon, and Richie Davidson

Narrator:

Aw, looks like you and Jeremy didn't get nominated.

That's okay though, the spring dance will still be fun!

Day of the dance: You and your friends all look stunning! Since you guys arrived at the dance with your dates you've been dancing and eating. This is the best night you've ever had at Eastwood High!

The rest of the night was a blast! Let's look at some pictures!

Here are some pictures from the night.

Play again for a better outcome!

<u>Worst Outcome:</u>

One week before the spring dance:

Principle on the intercom:

Hello everyone, I'd like to announce those in the running for the titles of spring queen and spring king.

For spring queen: Julia Fey, Ellen Kimmel, and Tina O'brien

For spring king: Jimmy Degenerees, Conan Fallon, and Richie Davidson

Narrator:

Aw, looks like you and Jeremy didn't get nominated.

That’s okay though, the spring dance will still be fun!

Day of the dance: Oh no! As your parents were driving you and Jeremy to the dance, they got a flat tire!

Luckily they had a spare tire, but you decided to get out of the car to watch them and fell in a puddle of mud.

You decide to just head to the dance anyway.

You and Jeremy are both about to walk into the dance auditorium but just before you do so, the principle stops Jeremy.

It turns out Jeremy is banned from the dance due to his fire alarm stunt.

You meet with your friends and try to enjoy the rest of the night but have a hard time doing so.

Here are some pictures from the night.

Play again for a better outcome!