INHERITANCE

"CLASSES ARE STRUCTS THAT CONTAIN FUNCTIONS"

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

RECAP

"CLASSES ARE STRUCTS THAT CONTAIN FUNCTIONS"

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

RECAP

“CLASSES ARE STRUCTS THAT CONTAIN FUNCTIONS”

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

RECAP

"CLASSES ARE STRUCTS THAT CONTAIN FUNCTIONS"

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

RECAP

"CLASSES ARE STRUCTS THAT CONTAIN FUNCTIONS"

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

RECAP

"CLASSES ARE STRUCTS THAT CONTAIN FUNCTIONS"

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

RECAP

"CLASSES ARE STRUCTS THAT CONTAIN FUNCTIONS"

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

AND COOLEST OF ALL, OTHER TYPES CAN "BUILD ON" THIS TYPE

RECAP

"CLASSES ARE STRUCTS THAT CONTAIN FUNCTIONS"

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

AND COOLEST OF ALL, OTHER TYPES CAN "BUILD ON" THIS TYPE

RECAP

“CLASSES ARE STRUCTS THAT CONTAIN FUNCTIONS”

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED TYPE

TAKEN TOGETHER, THESE IDEAS PRETTY MUCH DEFINE OBJECT ORIENTED PROGRAMMING

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

AND COOLEST OF ALL, OTHER TYPES CAN “BUILD ON” THIS TYPE

RECAP

TAKEN TOGETHER, THESE IDEAS PRETTY MUCH DEFINE OBJECT ORIENTED PROGRAMMING

SO LET'S CYCLE THROUGH THEM AGAIN (AND AGAIN)

"CLASSES ARE STRUCTS THAT CONTAIN FUNCTIONS"

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE MEANINGFUL TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED TYPE

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALIZATION, AND ALSO SOME CLEAN-UP

AND COOLEST OF ALL, OTHER TYPES CAN "BUILD ON" THIS TYPE

RECAP

TAKEN TOGETHER, THESE IDEAS PRETTY MUCH DEFINE OBJECT ORIENTED PROGRAMMING

SO LET'S CYCLE THROUGH THEM AGAIN (AND AGAIN)

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

AND COOLEST OF ALL, OTHER TYPES CAN "BUILD ON" THIS TYPE

RECAP

TAKEN TOGETHER, THESE IDEAS PRETTY MUCH DEFINE OBJECT ORIENTED PROGRAMMING

SO LET'S CYCLE THROUGH THEM AGAIN (AND AGAIN)

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

AND COOLEST OF ALL, OTHER TYPES CAN "BUILD ON" THIS TYPE

RECAP

TAKEN TOGETHER, THESE IDEAS PRETTY MUCH DEFINE OBJECT ORIENTED PROGRAMMING

SO LET'S CYCLE THROUGH THEM AGAIN (AND AGAIN)

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

AND COOLEST OF ALL, OTHER TYPES CAN "BUILD ON" THIS TYPE

RECAP

TAKEN TOGETHER, THESE IDEAS PRETTY MUCH DEFINE OBJECT ORIENTED PROGRAMMING

SO LET'S CYCLE THROUGH THEM AGAIN (AND AGAIN)

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED TYPE

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

AND COOLEST OF ALL, OTHER TYPES CAN "BUILD ON" THIS TYPE

RECAP

TAKEN TOGETHER, THESE IDEAS PRETTY MUCH DEFINE OBJECT ORIENTED PROGRAMMING

SO LET'S CYCLE THROUGH THEM AGAIN (AND AGAIN)

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

AND COOLEST OF ALL, OTHER TYPES CAN "BUILD ON" THIS TYPE

RECAP

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

THIS USER DEFINED TYPE IS CALLED THE CLASS, AND IT CORRESPONDS EXACTLY TO A STRUCT IN C

C++ GOES FAR BEYOND C IN MAKING USER-DEFINED CLASSES FIRST CLASS TYPES, ON PAR WITH THE SYSTEM TYPES SUCH AS INT, FLOAT ETC

RECAP

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

THIS USER DEFINED TYPE IS CALLED THE CLASS, AND IT CORRESPONDS EXACTLY TO A STRUCT IN C

C++ GOES FAR BEYOND C IN MAKING USER-DEFINED CLASSES FIRST CLASS TYPES, ON PAR WITH THE SYSTEM TYPES SUCH AS INT, FLOAT ETC

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

RECAP

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

THIS USER DEFINED TYPE IS CALLED THE CLASS, AND IT CORRESPONDS EXACTLY TO A STRUCT IN C

C++ GOES FAR BEYOND C IN MAKING USER-DEFINED CLASSES FIRST CLASS TYPES, ON PAR WITH THE SYSTEM TYPES SUCH AS INT, FLOAT ETC

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

A VARIABLE OF THIS CLASS IS CALLED AN OBJECT OF (OR AN INSTANCE OF) THE CLASS

RECAP

A VARIABLE OF THIS CLASS IS CALLED AN OBJECT OF (OR AN INSTANCE OF) THE CLASS

A CLASS IS BASICALLY A STRUCT ON STEROIDS

AN OBJECT IS A VARIABLE OF THAT CLASS

"OBJECT" AND "CLASS" ARE POSSIBLY THE 2 MOST IMPORTANT WORDS IN PROGRAMMING

RECAP

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

THIS USER DEFINED TYPE IS CALLED THE CLASS, AND IT CORRESPONDS EXACTLY TO A STRUCT IN C

C++ GOES FAR BEYOND C IN MAKING USER-DEFINED CLASSES FIRST CLASS TYPES, ON PAR WITH THE SYSTEM TYPES SUCH AS INT, FLOAT ETC

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

A VARIABLE OF THIS CLASS IS CALLED AN OBJECT OF (OR AN INSTANCE OF) THE CLASS

RECAP

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

THIS USER DEFINED TYPE IS CALLED THE CLASS, AND IT CORRESPONDS EXACTLY TO A STRUCT IN C

C++ GOES FAR BEYOND C IN MAKING USER-DEFINED CLASSES FIRST CLASS TYPES, ON PAR WITH THE SYSTEM TYPES SUCH AS INT, FLOAT ETC

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

A VARIABLE OF THIS CLASS IS CALLED AN OBJECT OF (OR AN INSTANCE OF) THE CLASS

RECAP

TAKEN TOGETHER, THESE IDEAS PRETTY MUCH DEFINE OBJECT ORIENTED PROGRAMMING

SO LET'S CYCLE THROUGH THEM AGAIN (AND AGAIN)

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

AND COOLEST OF ALL, OTHER TYPES CAN "BUILD ON" THIS TYPE

RECAP

TAKEN TOGETHER, THESE IDEAS PRETTY MUCH DEFINE OBJECT ORIENTED PROGRAMMING

SO LET'S CYCLE THROUGH THEM AGAIN (AND AGAIN)

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

AND COOLEST OF ALL, OTHER TYPES CAN "BUILD ON" THIS TYPE

RECAP

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

THE CONSTRUCTOR

A MEMBER FUNCTION TO TAKE CARE OF TAKING THESE IN, AND ASSIGNING TO THE CORRESPONDING MEMBER VARIABLES

RECAP

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

THE CONSTRUCTOR ALWAYS HAS THE SAME NAME AS THE CLASS

THE CONSTRUCTOR

A MEMBER FUNCTION TO TAKE CARE OF TAKING THESE IN, AND ASSIGNING TO THE CORRESPONDING MEMBER VARIABLES

RECAP

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

THE CONSTRUCTOR

A MEMBER FUNCTION TO TAKE CARE OF TAKING THESE IN, AND ASSIGNING TO THE CORRESPONDING MEMBER VARIABLES

THE CONSTRUCTOR ALWAYS HAS THE SAME NAME AS THE CLASS

THE C++ COMPILER WILL AUTOMATICALLY CALL THE CONSTRUCTOR WHEN AN OBJECT OF THE CLASS IS INSTANTIATED

RECAP

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

THE CONSTRUCTOR

A MEMBER FUNCTION TO TAKE CARE OF TAKING THESE IN, AND ASSIGNING TO THE CORRESPONDING MEMBER VARIABLES

THE DESTRUCTOR

A MEMBER FUNCTION TO TAKE CARE OF CLEANING UP THE OBJECT JUST BEFORE IT CEASES TO EXIST

THE DESTRUCTOR ALWAYS HAS THE SAME NAME AS THE CLASS, PRECEDED BY A TILDE (~)

RECAP

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

THE DESTRUCTOR

A MEMBER FUNCTION TO TAKE CARE OF CLEANING UP THE OBJECT JUST BEFORE IT CEASES TO EXIST

THE DESTRUCTOR ALWAYS HAS THE SAME NAME AS THE CLASS, PRECEDED BY A TILDE (~)

THE C++ COMPILER WILL AUTOMATICALLY CALL THE DESTRUCTOR WHEN AN OBJECT OF THE CLASS IS ABOUT TO CEASE TO EXIST

RECAP

TAKEN TOGETHER, THESE IDEAS PRETTY MUCH DEFINE OBJECT ORIENTED PROGRAMMING

SO LET'S CYCLE THROUGH THEM AGAIN (AND AGAIN)

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

AND COOLEST OF ALL, OTHER TYPES CAN "BUILD ON" THIS TYPE

RECAP

TAKEN TOGETHER, THESE IDEAS PRETTY MUCH DEFINE OBJECT ORIENTED PROGRAMMING

SO LET'S CYCLE THROUGH THEM AGAIN (AND AGAIN)

OFTEN, SOME VARIABLES AND FUNCTIONS JUST MAKE SENSE GROUPED TOGETHER

SOME OF THESE VARIABLES AND FUNCTIONS MIGHT BE INTERESTING TO THE REST OF THE PROGRAM, AND THESE SHOULD BE PUBLIC

OTHERS MIGHT BE INTERNAL PLUMBING, AND THESE SHOULD BE PRIVATE

THIS GROUP OF VARIABLES & FUNCTIONS IS EFFECTIVELY A NEW, USER-DEFINED, TYPE

ANYONE CAN CREATE A VARIABLE OF THIS TYPE

SUCH A VARIABLE MIGHT REQUIRE SOME INITIALISATION, AND ONCE ITS DONE, SOME CLEAN-UP

AND COOLEST OF ALL, OTHER TYPES CAN "BUILD ON" THIS TYPE

RECAP

AND COOLEST OF ALL, OTHER TYPES CAN "BUILD ON" THIS TYPE

A CLASS CAN 'INHERIT' FROM ANOTHER CLASS - START WITH ALL OF ITS MEMBERS AND METHODS, AND THEN BUILD ON IT

RECAP

AND COOLEST OF ALL, OTHER TYPES CAN "BUILD ON" THIS TYPE

A CLASS CAN 'INHERIT' FROM ANOTHER CLASS - START WITH ALL OF ITS MEMBERS AND METHODS, AND THEN BUILD ON IT

WE WILL GET BACK TO INHERITANCE ONCE WE REALLY UNDERSTAND OBJECTS AND CLASSES THOROUGHLY

RECAP

INHERITANCE

AND COOLEST OF ALL, OTHER TYPES CAN “BUILD ON” THIS TYPE

A CLASS CAN ‘INHERIT’ FROM ANOTHER CLASS - START WITH ALL OF ITS MEMBERS AND METHODS, AND THEN BUILD ON IT

WE WILL GET BACK TO INHERITANCE ONCE WE REALLY UNDERSTAND OBJECTS AND CLASSES THOROUGHLY

ITS TIME! LET’S PLUNGE INTO INHERITANCE!

BASICS OF INHERITANCE: LET'S GET ONE CLASS TO INHERIT FROM ANOTHER

BASICS OF INHERITANCE: LET'S GET ONE CLASS TO INHERIT FROM ANOTHER

WE HAVE A CLASS REPRESENTING A SHAPE

WE NOW NEED TO CREATE A CLASS REPRESENTING A CIRCLE

CLEARLY A CIRCLE IS-A SHAPE

CLEARLY A CIRCLE IS-A SHAPE

WE HAVE A CLASS REPRESENTING A SHAPE

WE NOW NEED TO CREATE A CLASS REPRESENTING A CIRCLE

INHERITANCE IS PERFECT FOR MODELLING IS-A RELATIONSHIPS

INHERITANCE IS PERFECT FOR MODELLING IS-A RELATIONSHIPS

WE HAVE A CLASS REPRESENTING A SHAPE

WE NOW NEED TO CREATE A CLASS REPRESENTING A CIRCLE

CLEARLY A CIRCLE IS-A SHAPE

THE CIRCLE CLASS SHOULD "INHERIT FROM" THE SHAPE CLASS

IS-A RELATIONSHIP

REAL-WORLD RELATIONSHIP

WE HAVE A CLASS REPRESENTING A SHAPE

WE NOW NEED TO CREATE A CLASS REPRESENTING A CIRCLE

CODE RELATIONSHIP

CLEARLY A CIRCLE IS A SHAPE

THE CIRCLE CLASS SHOULD “INHERIT FROM” THE SHAPE CLASS

THE CIRCLE CLASS SHOULD "INHERIT FROM" THE SHAPE CLASS

WHAT EXACTLY DOES THIS MEAN?

EVERY OBJECT OF THE CIRCLE CLASS WILL HAVE INSIDE IT AN OBJECT OF THE SHAPE CLASS

THE CIRCLE CLASS SHOULD "INHERIT FROM" THE SHAPE CLASS

WHAT EXACTLY DOES THIS MEAN?

EVERY OBJECT OF THE CIRCLE CLASS WILL HAVE INSIDE IT AN OBJECT OF THE SHAPE CLASS

WHEN THE CIRCLE IS BEING CONSTRUCTED/ DESTRUCTED, THE SHAPE OBJECT NEEDS TO BE CONSTRUCTED/DESTRUCTED TOO

THE CIRCLE CLASS SHOULD "INHERIT FROM" THE SHAPE CLASS

WHAT EXACTLY DOES THIS MEAN?

EVERY OBJECT OF THE CIRCLE CLASS WILL HAVE INSIDE IT AN OBJECT OF THE SHAPE CLASS

WHEN THE CIRCLE IS BEING CONSTRUCTED/DESTRUCTED, THE SHAPE OBJECT NEEDS TO BE CONSTRUCTED/DESTRUCTED TOO

THE CIRCLE OBJECT CAN DIRECTLY ACCESS SOME (NOT ALL) OF ITS INNER SHAPE OBJECT

THE CIRCLE CLASS SHOULD "INHERIT FROM" THE SHAPE CLASS

WHAT EXACTLY DOES THIS MEAN?

EVERY OBJECT OF THE CIRCLE CLASS WILL HAVE INSIDE IT AN OBJECT OF THE SHAPE CLASS

WHEN THE CIRCLE IS BEING CONSTRUCTED/DESTRUCTED, THE SHAPE OBJECT NEEDS TO BE CONSTRUCTED/DESTRUCTED TOO

THE CIRCLE OBJECT CAN DIRECTLY ACCESS SOME (NOT ALL) OF ITS INNER SHAPE OBJECT

THE CIRCLE CLASS SHOULD “INHERIT FROM” THE SHAPE CLASS

WHAT EXACTLY DOES THIS MEAN?

THERE IS A LOT GOING ON - LET’S TAKE IT ONE AT A TIME

EVERY OBJECT OF THE CIRCLE CLASS WILL HAVE INSIDE IT AN OBJECT OF THE SHAPE CLASS

WHEN THE CIRCLE IS BEING CONSTRUCTED/DESTRUCTED, THE SHAPE OBJECT NEEDS TO BE CONSTRUCTED/DESTRUCTED TOO

THE CIRCLE OBJECT CAN DIRECTLY ACCESS SOME (NOT ALL) OF ITS INNER SHAPE OBJECT

THE CIRCLE CLASS SHOULD "INHERIT FROM" THE SHAPE CLASS

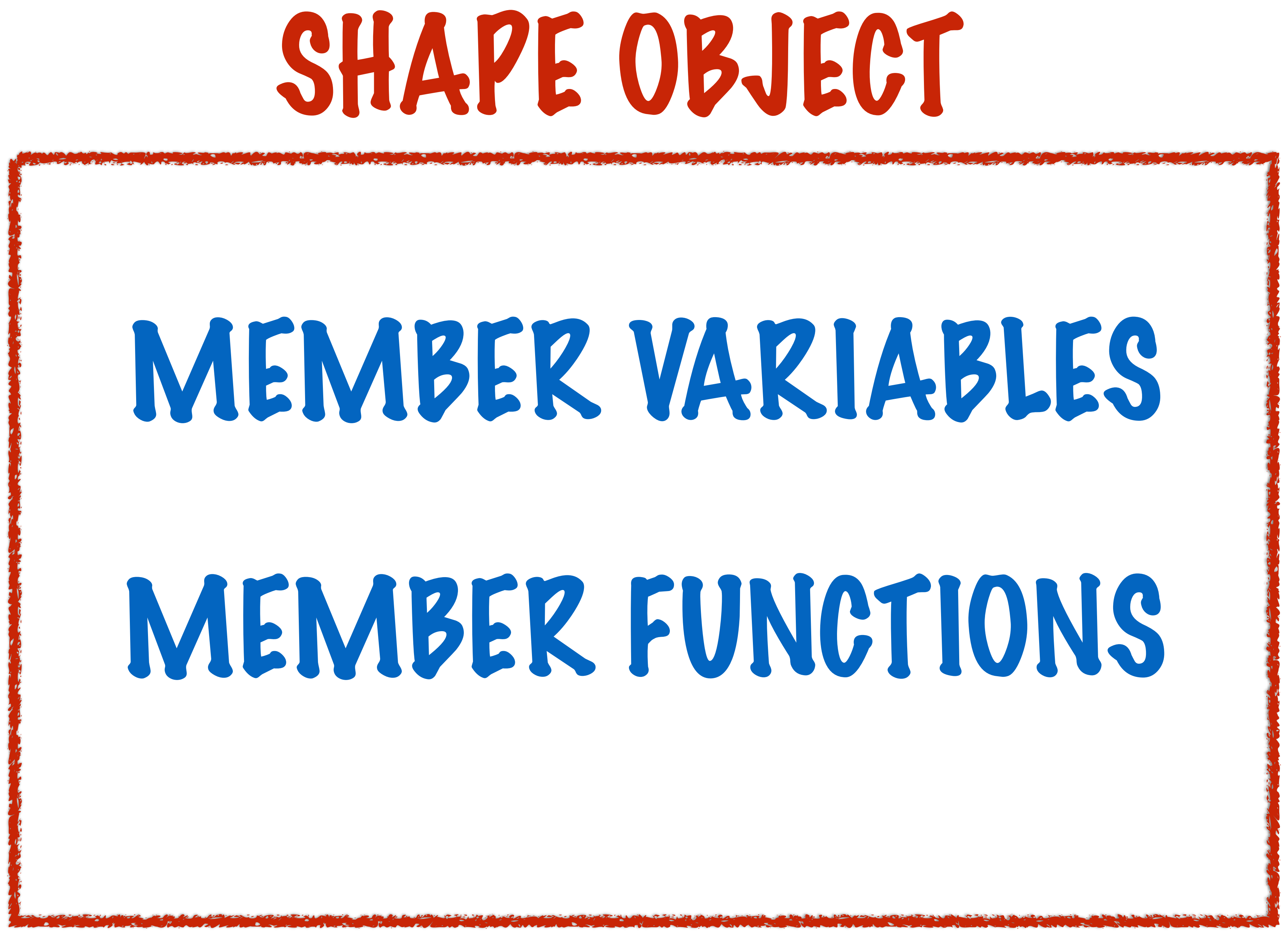
WHAT EXACTLY DOES THIS MEAN?

EVERY OBJECT OF THE CIRCLE CLASS WILL HAVE INSIDE IT AN OBJECT OF THE SHAPE CLASS

WHEN THE CIRCLE IS BEING CONSTRUCTED/DESTRUCTED, THE SHAPE OBJECT NEEDS TO BE CONSTRUCTED/DESTRUCTED TOO

THE CIRCLE OBJECT CAN DIRECTLY ACCESS SOME (NOT ALL) OF ITS INNER SHAPE OBJECT

EVERY OBJECT OF THE CIRCLE CLASS WILL HAVE INSIDE IT AN OBJECT OF THE SHAPE CLASS

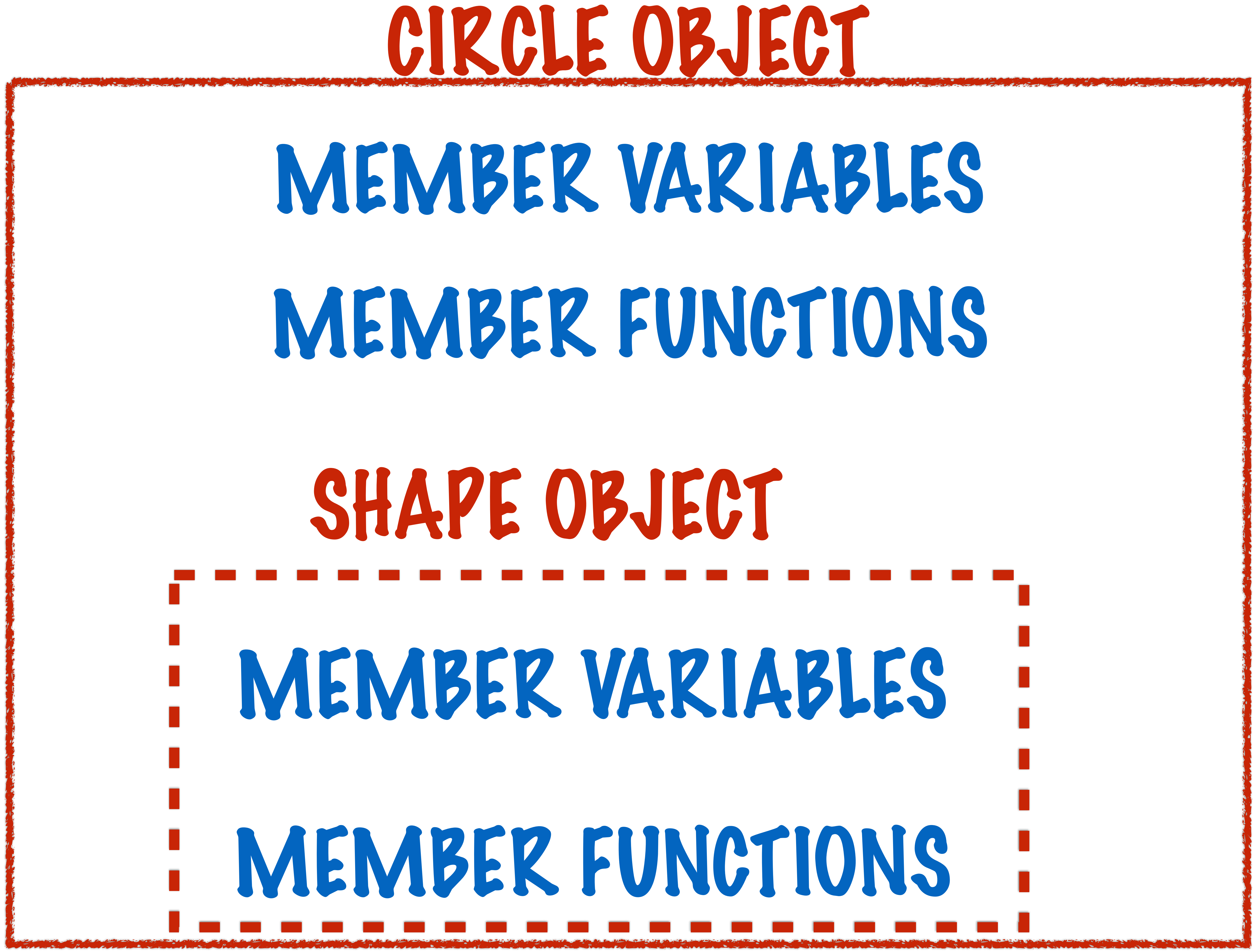


SHAPE OBJECT

MEMBER VARIABLES

MEMBER FUNCTIONS

EVERY OBJECT OF THE CIRCLE CLASS WILL HAVE INSIDE IT AN OBJECT OF THE SHAPE CLASS



CIRCLE OBJECT

MEMBER VARIABLES

MEMBER FUNCTIONS

SHAPE OBJECT

MEMBER VARIABLES

MEMBER FUNCTIONS

THE CIRCLE CLASS SHOULD "INHERIT FROM" THE SHAPE CLASS

WHAT EXACTLY DOES THIS MEAN?

EVERY OBJECT OF THE CIRCLE CLASS WILL HAVE INSIDE IT AN OBJECT OF THE SHAPE CLASS

WHEN THE CIRCLE IS BEING CONSTRUCTED/DESTRUCTED, THE SHAPE OBJECT NEEDS TO BE CONSTRUCTED/DESTRUCTED TOO

THE CIRCLE OBJECT CAN DIRECTLY ACCESS SOME (NOT ALL) OF ITS INNER SHAPE OBJECT

THE CIRCLE CLASS SHOULD "INHERIT FROM" THE SHAPE CLASS

WHAT EXACTLY DOES THIS MEAN?

EVERY OBJECT OF THE CIRCLE CLASS WILL HAVE INSIDE IT AN OBJECT OF THE SHAPE CLASS

WHEN THE CIRCLE IS BEING CONSTRUCTED/DESTRUCTED, THE SHAPE OBJECT NEEDS TO BE CONSTRUCTED/DESTRUCTED TOO

THE CIRCLE OBJECT CAN DIRECTLY ACCESS SOME (NOT ALL) OF ITS INNER SHAPE OBJECT

WHEN THE CIRCLE IS BEING CONSTRUCTED/DESTRUCTED, THE SHAPE OBJECT NEEDS TO BE CONSTRUCTED/DESTRUCTED TOO

DURING CONSTRUCTION OF THE CIRCLE..

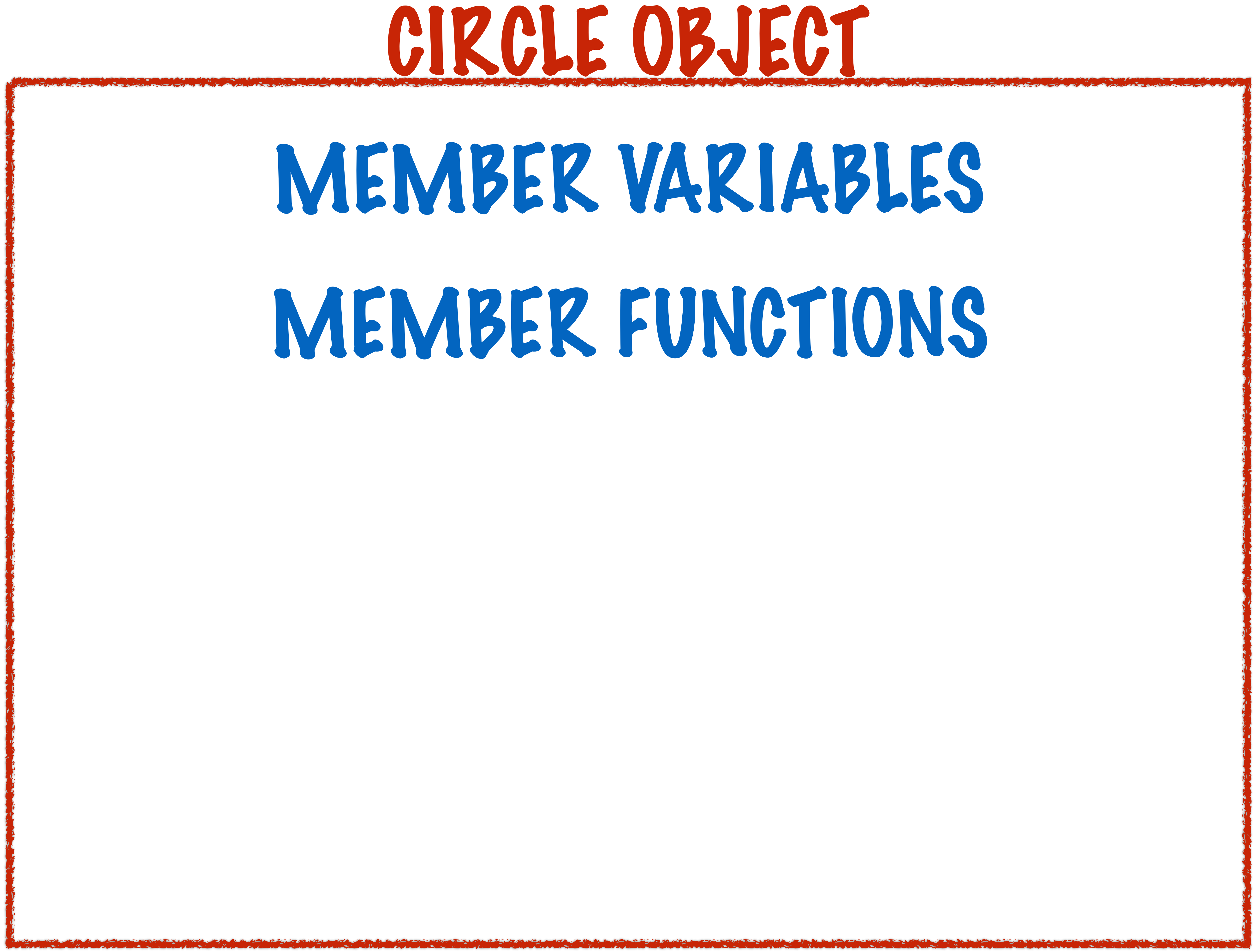
THE CONSTRUCTOR OF THE SHAPE IS IMPLICITLY CALLED FROM THE CONSTRUCTOR OF THE CIRCLE

DURING DESTRUCTION OF THE CIRCLE..

THE DESTRUCTOR OF THE SHAPE IS IMPLICITLY CALLED FROM THE DESTRUCTOR OF THE CIRCLE

DURING CONSTRUCTION OF THE CIRCLE..

THE CONSTRUCTOR OF THE SHAPE IS IMPLICITLY CALLED FROM THE CONSTRUCTOR OF THE CIRCLE



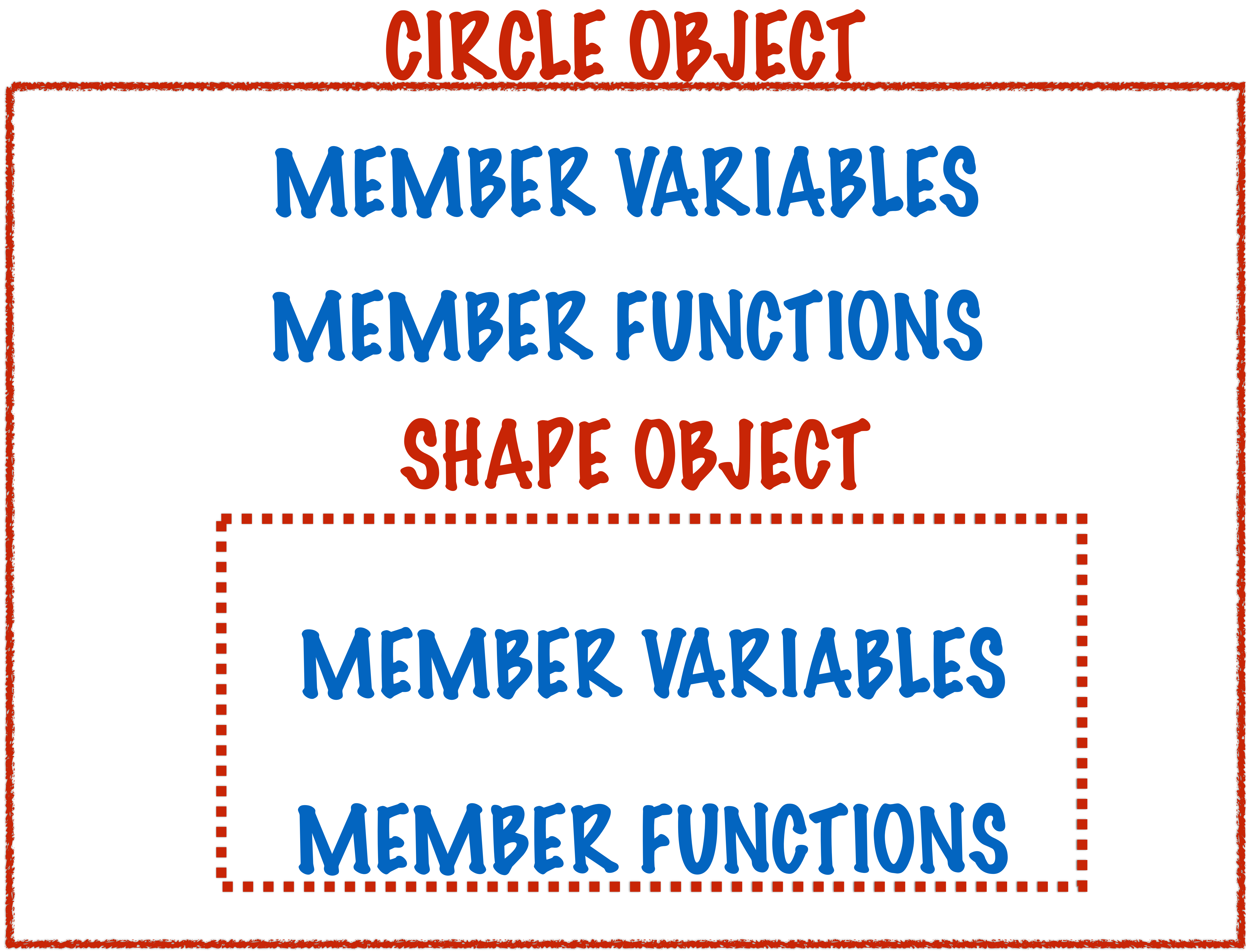
CIRCLE OBJECT

MEMBER VARIABLES

MEMBER FUNCTIONS

DURING CONSTRUCTION OF THE CIRCLE..

THE CONSTRUCTOR OF THE SHAPE IS IMPLICITLY CALLED FROM THE CONSTRUCTOR OF THE CIRCLE



CIRCLE OBJECT

MEMBER VARIABLES

MEMBER FUNCTIONS

SHAPE OBJECT

MEMBER VARIABLES

MEMBER FUNCTIONS

WHEN THE CIRCLE IS BEING CONSTRUCTED/DESTRUCTED, THE SHAPE OBJECT NEEDS TO BE CONSTRUCTED/DESTRUCTED TOO

DURING CONSTRUCTION OF THE CIRCLE..

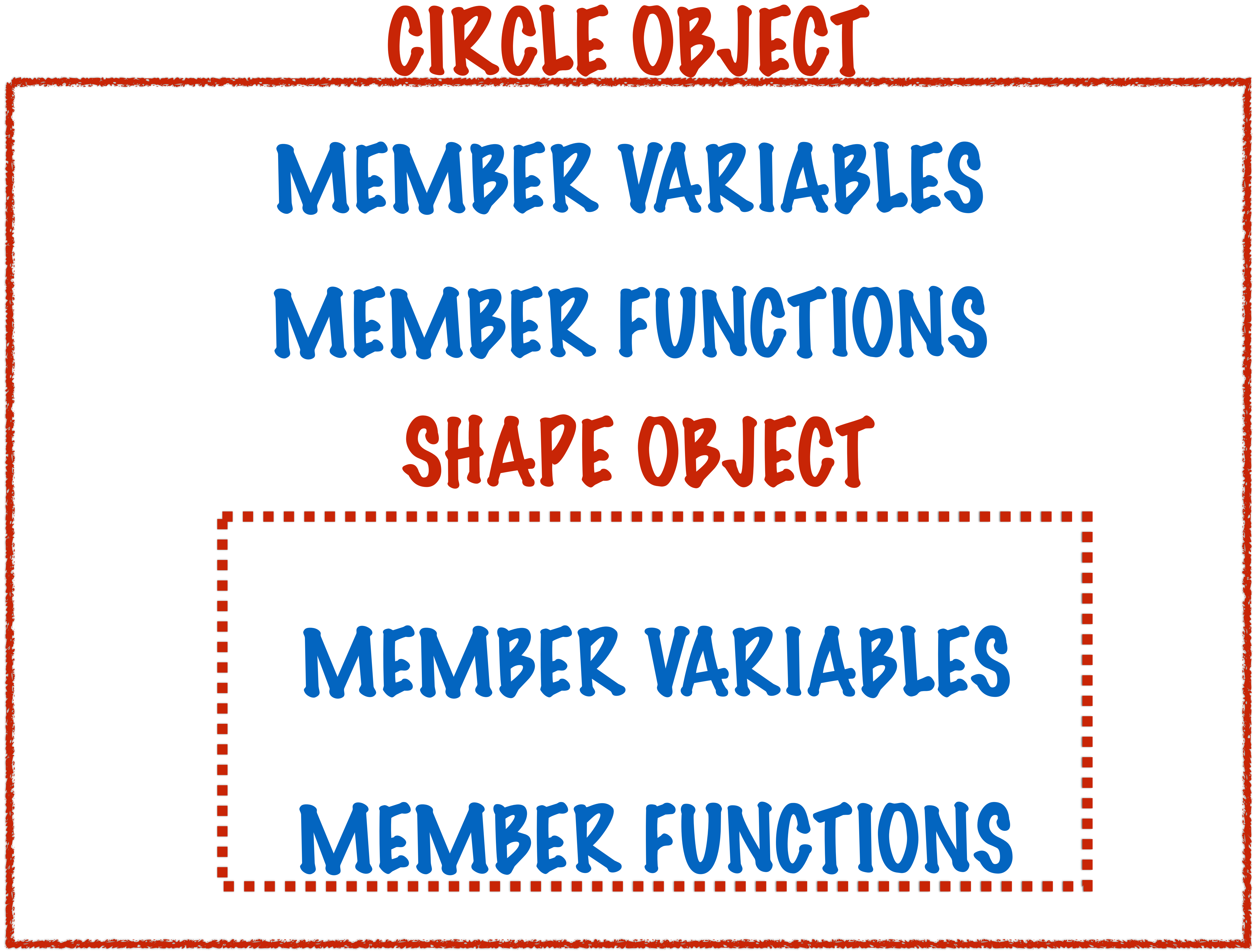
THE CONSTRUCTOR OF THE SHAPE IS IMPLICITLY CALLED FROM THE CONSTRUCTOR OF THE CIRCLE

DURING DESTRUCTION OF THE CIRCLE..

THE DESTRUCTOR OF THE SHAPE IS IMPLICITLY CALLED FROM THE DESTRUCTOR OF THE CIRCLE

DURING DESTRUCTION OF THE CIRCLE..

THE DESTRUCTOR OF THE SHAPE IS IMPLICITLY
CALLED FROM THE DESTRUCTOR OF THE CIRCLE



CIRCLE OBJECT

MEMBER VARIABLES

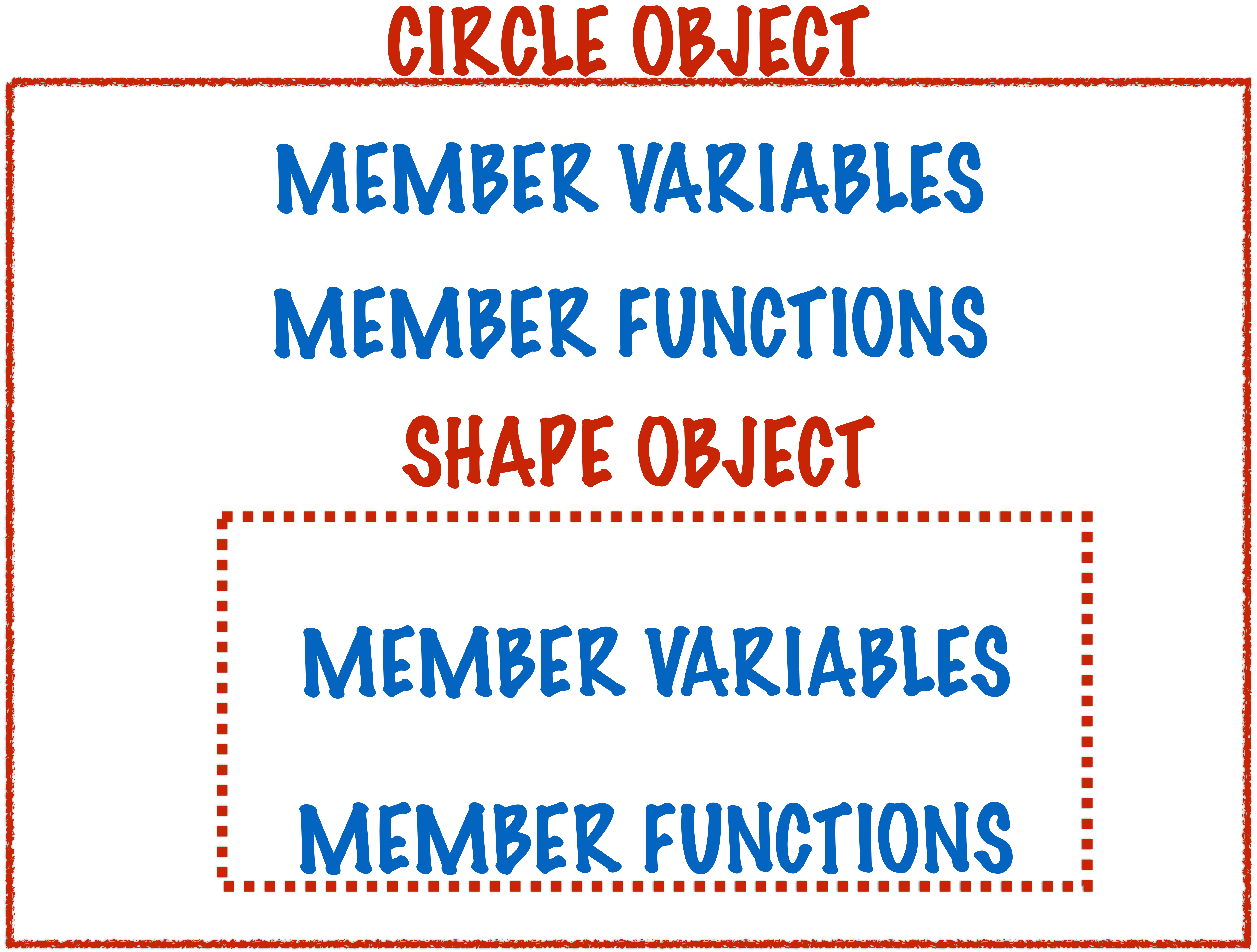
MEMBER FUNCTIONS

SHAPE OBJECT

MEMBER VARIABLES

MEMBER FUNCTIONS

"BASE CLASS" AND "DERIVED CLASS" ARE THE TECHNICAL TERMS FOR SHAPE AND THE CIRCLE RESPECTIVELY



CIRCLE OBJECT

MEMBER VARIABLES

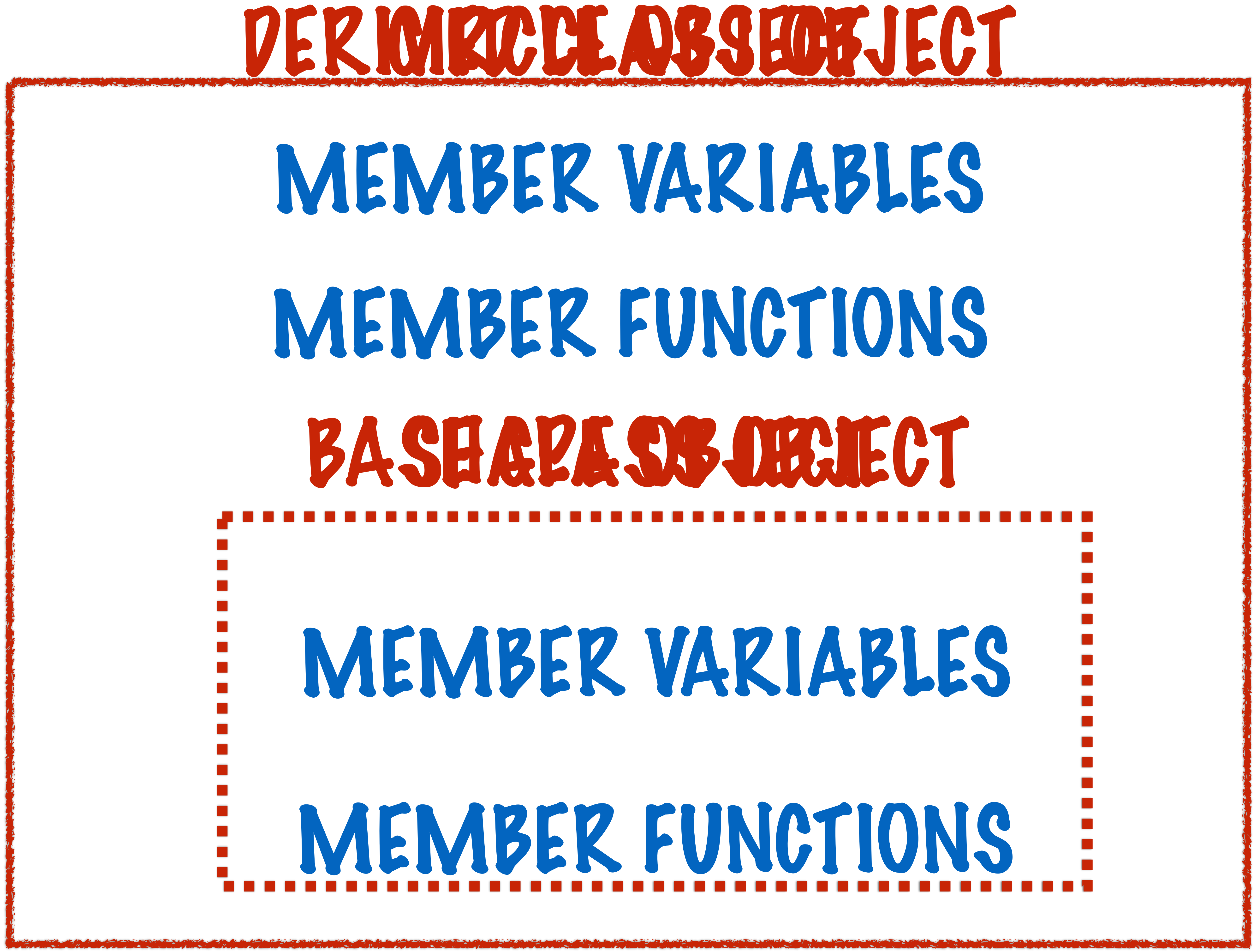
MEMBER FUNCTIONS

SHAPE OBJECT

MEMBER VARIABLES

MEMBER FUNCTIONS

"BASE CLASS" AND "DERIVED CLASS" ARE THE TECHNICAL TERMS FOR SHAPE AND THE CIRCLE RESPECTIVELY



DERIVED CLASS OBJECT

CIRCLE OBJECT

MEMBER VARIABLES

MEMBER FUNCTIONS

BASE CLASS OBJECT

SHAPE OBJECT

MEMBER VARIABLES

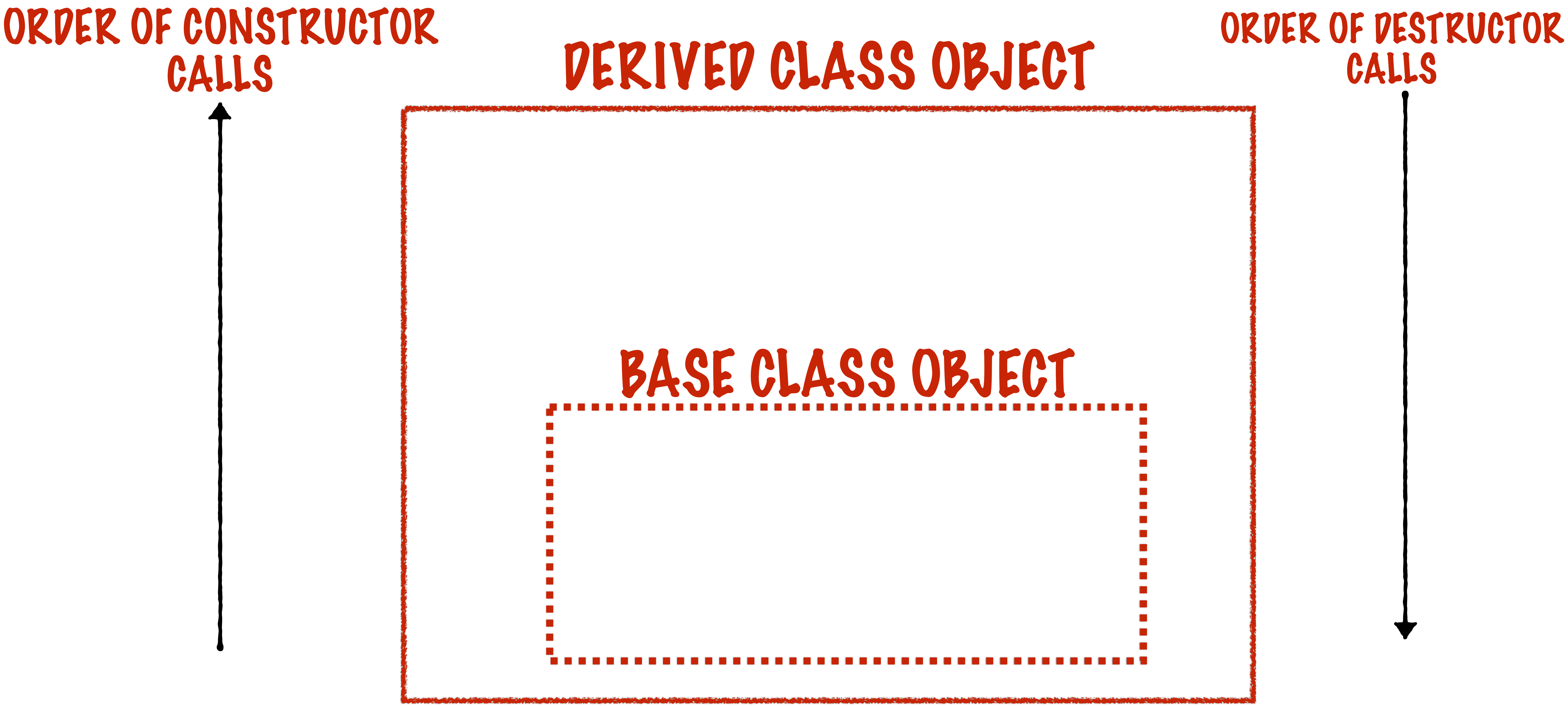
MEMBER FUNCTIONS

“BASE CLASS” AND “DERIVED CLASS”

REMEMBER THESE TERMS,
THEY ARE VERY IMPORTANT!

"BASE CLASS" AND "DERIVED CLASS"

REMEMBER THESE TERMS, THEY ARE VERY IMPORTANT!



WHEN THE CIRCLE IS BEING CONSTRUCTED/DESTRUCTED, THE SHAPE OBJECT NEEDS TO BE CONSTRUCTED/DESTRUCTED TOO

DURING CONSTRUCTION OF THE CIRCLE..

BASE CLASS CONSTRUCTOR IS CALLED BEFORE THE DERIVED CLASS CONSTRUCTOR

DURING DESTRUCTION OF THE CIRCLE..

BASE CLASS DESTRUCTOR IS CALLED AFTER THE DERIVED CLASS DESTRUCTOR

THE CIRCLE CLASS SHOULD "INHERIT FROM" THE SHAPE CLASS

WHAT EXACTLY DOES THIS MEAN?

EVERY OBJECT OF THE CIRCLE CLASS WILL HAVE INSIDE IT AN OBJECT OF THE SHAPE CLASS

WHEN THE CIRCLE IS BEING CONSTRUCTED/DESTRUCTED, THE SHAPE OBJECT NEEDS TO BE CONSTRUCTED/DESTRUCTED TOO

THE CIRCLE OBJECT CAN DIRECTLY ACCESS SOME (NOT ALL) OF ITS INNER SHAPE OBJECT

THE CIRCLE CLASS SHOULD "INHERIT FROM" THE SHAPE CLASS

WHAT EXACTLY DOES THIS MEAN?

EVERY OBJECT OF THE CIRCLE CLASS WILL HAVE INSIDE IT AN OBJECT OF THE SHAPE CLASS

WHEN THE CIRCLE IS BEING CONSTRUCTED/DESTRUCTED, THE SHAPE OBJECT NEEDS TO BE CONSTRUCTED/DESTRUCTED TOO

THE CIRCLE OBJECT CAN DIRECTLY ACCESS SOME (NOT ALL) OF ITS INNER SHAPE OBJECT

THE CIRCLE OBJECT CAN DIRECTLY ACCESS SOME (NOT ALL) OF ITS INNER SHAPE OBJECT

THERE ARE 3 LEVELS OF ACCESS: PUBLIC, PRIVATE, PROTECTED

THERE ARE 3 LEVELS OF ACCESS: PUBLIC, PRIVATE, PROTECTED

PUBLIC: ACCESSIBLE EVERYWHERE, EXTERNALLY AS WELL AS IN DERIVED CLASS

PRIVATE: NOT ACCESSIBLE ANYWHERE EXTERNALLY, NOT EVEN IN DERIVED CLASS

PROTECTED: ACCESSIBLE ONLY TO DERIVED CLASSES, NOWHERE ELSE EXTERNALLY

THERE ARE 3 LEVELS OF ACCESS: PUBLIC, PRIVATE, PROTECTED

PUBLIC: ACCESSIBLE EVERYWHERE, EXTERNALLY AS WELL AS IN DERIVED CLASS

PRIVATE: NOT ACCESSIBLE ANYWHERE EXTERNALLY, NOT EVEN IN DERIVED CLASS

PROTECTED: ACCESSIBLE ONLY TO DERIVED CLASSES, NOWHERE ELSE EXTERNALLY

THERE ARE ALSO 3 TYPES OF INHERITANCE: PUBLIC, PRIVATE, PROTECTED

THERE ARE 3 LEVELS OF ACCESS: PUBLIC, PRIVATE, PROTECTED

SPECIFIED WITHIN A CLASS

GOVERN HOW BASE CLASS MEMBERS ARE ACCESSED IN DIRECTLY DERIVED CLASSES

THERE ARE ALSO 3 TYPES OF INHERITANCE: PUBLIC, PRIVATE, PROTECTED

SPECIFIED BETWEEN A BASE AND DERIVED CLASS

GOVERN HOW BASE CLASS MEMBERS ARE ACCESSED IN INDIRECTLY DERIVED CLASSES