3. Class Creation

Starter Quiz ·

AP Computer Science A Name: _____ Score: _____

1. Abstraction is best described as...

   ☐ deleting comments

   ☐ hiding details to focus on the main idea

   ☐ a run-time error

   ☐ writing more code

2. A modular design (independent, well-named pieces) is mainly easier to...

   ☐ run without a computer

   ☐ compile only once

   ☐ hide from users

   ☐ test, maintain, and reuse

3. Which three parts organize the Student class design taught here?

   ☐ loops, ifs, casts

   ☐ public, private, static

   ☐ int, double, boolean

   ☐ instance variables, constructors, methods

4. A constructor has...

   ☐ any name you like

   ☐ an int return type

   ☐ a void return type

   ☐ the class's name and no return type

5. A method that returns information without changing the object is a(n)...

   ☐ accessor (getter)

   ☐ mutator (setter)

   ☐ cast

   ☐ constructor

6. A method can change the caller's int variable by modifying its parameter.

   ☐ True ☐ False

7. Inside the shown Student class, a static method may use return score; without specifying any Student object.

   ☐ True ☐ False

8. A parameter with the same name as an instance variable shadows (hides) that instance variable inside the method.

   ☐ True ☐ False

9. The this keyword can be used inside a static method.

   ☐ True ☐ False

10. Breaking a big problem into smaller named behaviors (methods) is called method ____ (one word).

    _____

3. Class Creation

Answers ·

AP Computer Science A

1. **hiding details to focus on the main idea**

   Abstraction hides how something works so you can use what it does.

2. **test, maintain, and reuse**

   Modularity makes changes safer and pieces reusable.

3. **instance variables, constructors, methods**

   Instance fields hold state, constructors initialize new objects, and methods provide behavior. Constructors are not formally class members in Java terminology.

4. **the class's name and no return type**

   Same name as the class, no return type – not even void.

5. **accessor (getter)**

   Accessors read and return; mutators change state.

6. **False**

   Primitives are copied – the caller's int is untouched.

7. **False**

   False: score is an instance field and this static method has no implicit receiver. With a non-null Student s parameter, return s.score; is valid inside Student.

8. **True**

   Inside the method, the name means the parameter; use this to reach the field.

9. **False**

   Static methods have no particular object, so no this.

10. **decomposition**

    Method decomposition splits work into focused methods.