AP Computer Science Principles lessons 1-12

Unit 5 Part 1 Review

Name: Kelly

Define the following terms:

- Callback function – a function specified as part of an event listener, it is written by the programmer but called by the system as the result of an event trigger
- Event – an action that causes something to happen
- Event-driven program – a program designed to run blocks of code or functions in response to specified events (e.g. a mouse click)
- Event handling – an overarching term for the coding tasks involved in making a program respond to events by triggering functions
- Event listener – a command that can be set up to trigger a function when a particular type of event occurs on a particular UI element
- UI elements – on-screen objects, like buttons, images, text boxes, pull down menus, screens and so on
- User Interface – The visual elements of a program through which a user controls or communicates in an application. Often & abbreviated UI
- Debugging – Finding and fixing problems in your algorithm or program

repeat - Event-driven program – a program designed to run blocks of code or functions in response to specified events (e.g. mouse click)

- Event handling – repeat

- Data Type – all values in a programming language have a "type" - such as a number, boolean, or string - that dictates how the computer will interpret it.
  ex: 7+5 is different than "7" + "5"
- Expression – any valid unit of code that ~~results to~~ resolves to a value

- Variable – a placeholder for a piece of information that can change

- == - the equality operator ("equal equal") is used to compare two values, and returns a Boolean (true/false). Avoid confusion with the assignment operator ("=")
- Global variable – a variable whose scope is "global" to the program, it can be used and updated by any part of the code. Its global scope is typically derived from a variable being declared (created) outside of any function, object, or method
- If-statement – the common programming structure that implements "conditional statements"

- Local variable – A variable with local scope is one that can only be seen, used, and updated by code within the same scope. Typically, this means the variable was declared (created) inside a function - includes function parameter variables
- Variable scope – dictates ~~the~~ what portions of the code can "see" or use a variable, typically derived from where the variable was first created.

- **Concatenate** – to link together or join. Typically used when joining together text strings in programming (e.g. "Hello" + "name")
- **String** – Any sequence of characters between quotation marks (ex. "hello", "42", "this is a string") → a data type
- **Conditionals** – Statements that only run under certain conditions

- **Selection** – A generic term for a type of programming statement (usually if-statement) that uses a Boolean condition to determine, or select, whether or not to run a certain block of statements
- **Boolean** – a single value of either True or False → can be a data type

- **Boolean Expression** – in programming, an expression that evaluates to True or False

- **Iterate** – to repeat in order to achieve, or get closer to, a desired goal (think loops)

- **While loop** – a programming construct used to repeat a set of commands (loop) as long as (while) a boolean condition is true.
- **Models and Simulations** – a program which replicates or mimics key features of a real world event in order to investigate its behavior w/o the cost, time, or danger of running an experiment in real life.

Short Answer:

1) What is the difference between logical errors and syntax errors?

logical: harder to solve, computer does not report error, code's behavior is different than desired
syntax: easy to solve, show up in the console, are related to format of the symbols in the code (ex: missing semicolon)

2) What is the difference between event handling and if statements?

events: set up by programmer, but triggered by computer at any moment in time (when)

if statement: a way a programmer can have her code make a decision during the normal flow of execution to do one thing or another (if)

3) When is it more helpful to use a global variable than a local variable?

Global variables are useful for keeping track of data over the lifetime of the program that's running

*usually created/declared at the top of code

4) Give an example of a situation where you might use an if-statement. Give another example of a situation where you might use an if-else-statement

*should have 2 examples

5) Describe in your own words how a while loop works

```
while (condition)
{
  // block of code
}
```

→ the code will run so long as the "condition" is true
→ the block of code will be repeated until the condition is false

6) Why do we use models and simulations to represent real life situations?

→ can save time, resources, or even danger of running an experiment in real life

7) Note: Make sure you review how to do math in coding
8) Note: One great way to study is to review concepts in Code Studio and actually look at and create code
   a. Example: What happens when you restructure code?