1.) The security advantages to assigning seperate uid's to each application are that individual apps cannot interact with each other, and that each app has limited access to the system resources. Both of these advantages reduce the ability of a maliciously programmed application from compromising the system as a whole.

2.)
- I agree with the statement. Consider a server with little to no integrity. Because of this lack of integrity, it would be easy to improperly modify information such as my access privilege, thus allowing me to access unauthorized information, which means that confidentiality is compromised, thus a server without integrity cannot have confidentiality.
- One could argue that it is impossible to provide integrity without confidentiality. Consider a server that completely lacks confidentiality, then all of the server's information would be freely available to all who wish to view it, including the root password. Malicious users would easily be able to retrieve this password and improperly modify information on the server, thus violating integrity. From this one can conclude that without confidentiality, it is impossible for a system to satisfy integrity.

3.) The principle of least privilege concerns the access that subjects get to objects. It states that "subjects should only have access to the objects needed to perform routine, authorized rights". This principle is important to computer security because it prohibits ordinary users from acting with higher privilege than they should have access to, preventing them from causing damage both inadvertently, and maliciously.

4.) The situation in which Carol changes the amount on Angelo's check is a clear violation of integrity, as Carol is improperly modifying the information on the check. One might be able to detect this forgery through forensics, or by analyzing the handwriting of the number.

5.) The code for accessCheck is given in C++ as follows

```
int accessCheck(unsigned int uid, unsigned int gid,
                unsigned int p, int f) {
    Permission permissions = getPermission(f);

    if (uid == permissions.uid) {
        return p | permissions.u == p ? 1 : 0;
    } else if (gid == permissions.gid) {
        return p | permissions.g == p ? 1 : 0;
    }

    return p | permissions.o == p ? 1 : 0;
}
```