# Programming Challenge 11

**1) Briefly define constructor as it relates to a C++ class. When are constructors called/invoked?**
A constructor is a special function which belongs to class. It is called automatically when a new object is instantiated. According to our textbook a constructor is "a member function that is automatically called when an object of that class is declared. A constructor is used to initialize the values of some or all memeber variables and to do any other sort of initialization that may be needed" (264, 3rd Edition).

**2) What are the two ways that constructors are different from other member functions of a C++ class?**
According to our textbook a constructor is defined in the same way as other member functions except that "a constructor must have the same name as the class" and "a constructor definition cannot return a value" (265, 3rd Edition).

**3) Can constructors be overloaded?**
Yes, constructors can be overloaded. According to our textbook a constructor can be overloaded just like any other function (267, 3rd Edition).

**4) Write a declaration for a default constructor for the following class.**

```
class Book
{
    private:
        string authorName;
        string title;
        unsigned int pages;
};

public:
        Book();
```

**5) Write a definition for a default constructor for the following class. In the body of the constructor, initialize to the values authorName("Unknown"), title("No title"), pages (0).**

```
class Book
{
    private:
        string authorName;
        string title;
        unsigned int pages;
};

Book::Book()
{
        authorName = "Unknown";
        title = "No title";
        pages = 0;
}
```

**6) Write a declaration for an overloaded constructor for the following class. The overloaded constructor must include parameters for all of the member variables of the class.**

```
class Book
{
    private:
        string authorName;
        string title;
        unsigned int pages;
};

public:
        Book( string authorNameValue, string titleValue, usigned int pagesValue );
```

**7) Write a definition for an overloaded constructor for the following class. Assign the values of the variables in the body of the function.**

```
class Book
{
    private:
        string authorName;
        string title;
        unsigned int pages;
};

Book::Book( string authorNameValue, string titleValue, usigned int pagesValue )
{
        authorName = authorNameValue;
        title = titleValue;
        pages = pagesValue;
}
```

**8) Write a definition for an overloaded constructor for the following class. Assign the values of the variables in the initialization section.**

```
class Book
{
    private:
        string authorName;
        string title;
        unsigned int pages;
};

Book::Book( string authorNameValue, string titleValue, usigned int pagesValue ) :
authorName( authorNameValue ), title( titleValue ), pages( pagesValue )
{       /* empty */     }
```

**9) Part 1: Write a declaration for an overloaded constructor for the following class. The overloaded constructor must have the following default arguments for its parameters: authorName ("Unknown"), title ("No title"), pages (0). Part 2: If this function is implemented, is it possible to also define a default constructor?**

```
class Book
{
    private:
        string authorName;
        string title;
        unsigned int pages;
};

public:
        Book( string authorNameValue = "Unknown", string titleValue = "No title", unsigned int pagesValue =
0 );
```

No, adding a default constructor when another constructor exists having every parameter set with a default value would cause a collision to occur during compilation.

**10) Can objects of the following class be created, without an explicit declaration/definition of a constructor?**

```
class Book
{
    private:
        string authorName;
        string title;
        unsigned int pages;
};
```

Yes.  According to our textbook "if you define a class and include absolutely no constructors of any kind, then a default constructor will be automatically created.  This default constructor does not do anything, but it does give you an uninitialized object of the class type, which can be assigned to a variable of the class type" (271, 3rd Edition).