

UAH Electrical and Computer Engineering Department
Project02 (5 points) Version 1.0

< Important Note >

All CPE 212 projects are automatically graded in order to provided you timely feedback. So, it is critical that you follow all directions for the preparation and submission of your projects for grading. Failure to follow the directions may result in zero credit (0 points).

<Project02 Goals>

As part of this assignment, you will once again practice implementing components that will be integrated to form a larger product. This time you will write several classes that will be integrated into a larger software product provided by the instructor.

<Project02 Description>

For this project, you will **complete** the provided partial C++ program that uses classes and inheritance to model **playing cards**.

The inheritance hierarchy is described by the UML class diagram on the following page. **RedCard** and **BlackCard** inherit from **Card**. **Heart** and **Diamond** inherit from **RedCard**. **Club** and **Spade** inherit from **BlackCard**.

The provided **Driver** program and input files create objects of each of the above data types, and these objects are used to test all member functions of each class for correct operation.

Each card is described by three private attributes that are defined in the **Card** class:

- a **value** (an **int** from **2-10** for number cards, **11-14** for face cards/aces, or **0** for unknown),
- a **color** (a **string** "**red**" or "**black**" or "**unknown**"), and
- a **suit** (a **char** '**H**' for heart, '**D**' for diamond, '**C**' for club, '**S**' for spade, or '**U**' for unknown).

The function **main()** has already been implemented for you to provide a common way for everyone to test their code. The function **main()** will scan a sample input file and perform the requested series of operations by invoking your class methods as needed.

It is very important that you closely follow all directions for this project since you will be working with a total of **fifteen source code files**.

Don't panic – many of these files contain only a few lines of code.

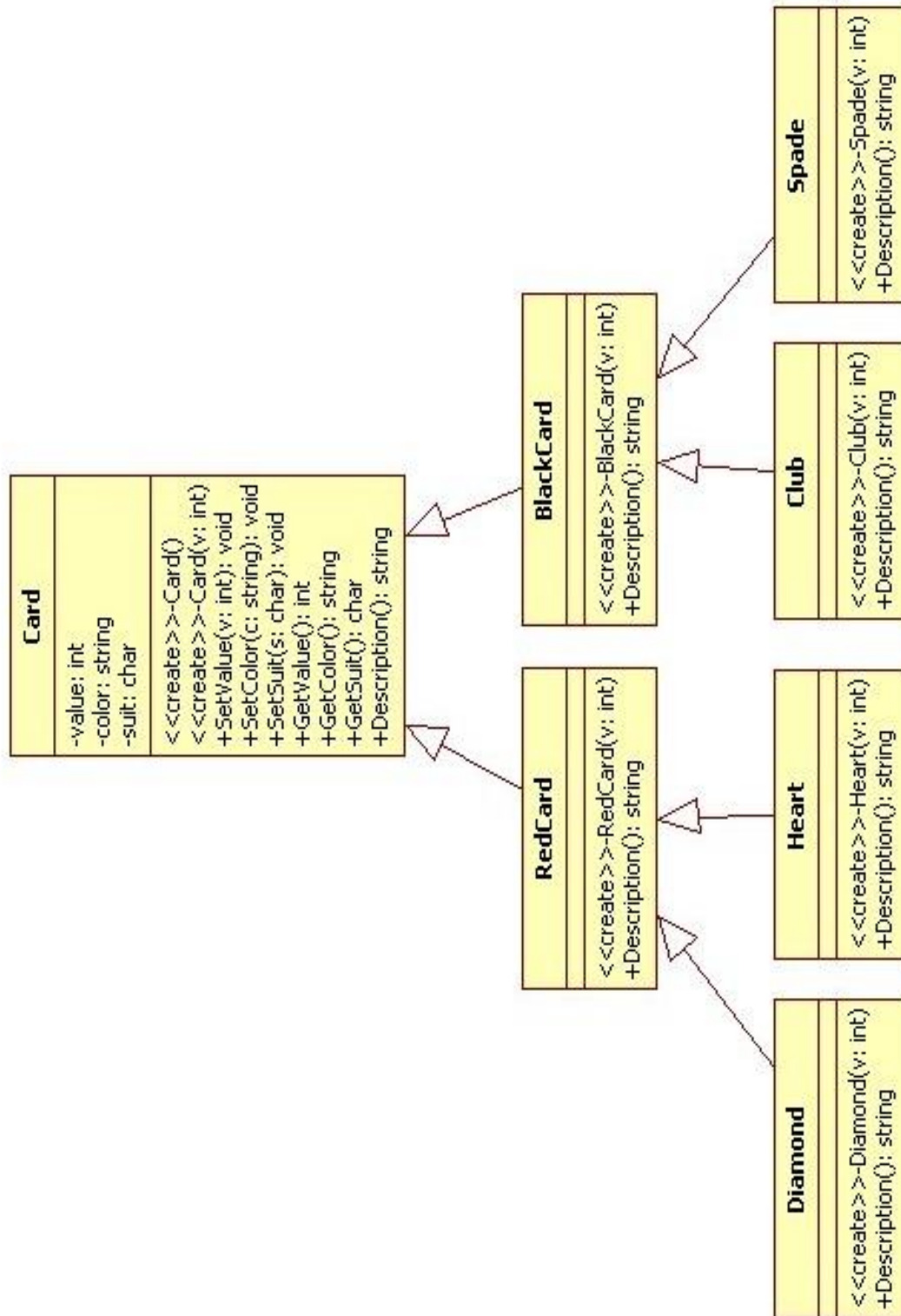
- **Do not modify main.cpp !!!**
- **All program output is performed by the code in the main.cpp file – do not include any output statements in the files you write!!!**

The interfaces to each of these functions you must write are described in detail on subsequent pages and in the files provided

Hint: Match the output of your program to that of the sample solution!!!

Directions for running the sample solution appear on the following page.

<Project02 UML Class Diagram>



+ indicates a public member

- indicates a private member

<<create>> indicates a constructor

Arrows point to the base class (parent). So, **Card** is the parent/grandparent of all other classes.

Running the Sample Solution on **blackhawk**

The best description of what your code should do is the Sample Solution for the project.

Run the sample solution by typing the following at **blackhawk** terminal window command prompt

```
/home/work/cpe212/project02/p02 inputfilename
```

where **inputfilename** is the name of one of the provided input files (for example, **p02input1.txt**).
Your current working directory must contain the input files for this to work.

Unzipping Sample Input Files on **blackhawk**

Use your browser browser to access Canvas and project02_materials.zip into your **Project02** directory. At terminal window prompt, use the **unzip** utility to uncompress the files.

For example: **unzip NameOfZipFileHere**

to unzip the files into your current directory.

Since this project is worth **five points**, you have been given five input files to test your program.

Running the Preview Script on **blackhawk**

Run the preview script by typing the following in a **blackhawk** terminal window command prompt

```
/home/work/cpe212data/project02/preview02.bash
```

This script will run both the Sample Solution AND your project02 executable program on the complete set of input files, and it compares the outputs of the two programs line by line to identify errors in your program's outputs. Make sure that the output of your program exactly matches the output of the Sample Solution.

<Project02 Compilation Instructions>

This project consists of **fifteen C++ files** so you do not want to compile and link these files manually. The **makefile** contains the sequence of commands required to compile and assemble (link) your executable program. The provided file works on **blackhawk**. So, for this assignment, all you must do to compile the program is to use the following command at the Linux command line

```
make
```

which will create an executable named **project02** from the provided files **main.cpp** and the files you have written.

If your program compiled successfully, you may then type

```
./project02 NameOfInputFile
```

to execute your program assuming that the input file is located in the same directory with the executable. (For example, **./project02 p02input1.txt**)

To force all files to be recompiled and relinked, type the following sequence of commands at the Linux prompt.

```
make clean
```

```
make
```

<Project02 Include File Constraints>

On Project02, all allowable include files are already specified in the provided source code materials.

Card header files such as “card.h”, “blackcard.h”, or “club.h” are allowed

No additional include files are allowed!!!

Failure to follow these directions will result in zero (0) credit on this assignment.

<Project02 Hints>

Before you can compile and test any code for this project, you must have a functional skeleton of the solution!!!

- Use the code appearing in **card.cpp** to complete the class declaration in **card.h** by adding the missing function prototypes to **card.h**
- Use the function prototypes appearing in the other header (specification) files to create skeleton class implementation files (for example, use **xyz.h** to create **xyz.cpp**) to form the program skeleton.

Hint: The provided **card.cpp** is an example of a skeleton!!

- Be sure that you can successfully use **make** to compile this skeleton solution before adding any additional code.

After you see that the skeleton compiles and runs, **start adding code at the TOP of the class diagram and work your way towards the bottom – ONE CLASS AT A TIME!!**

- The comments appearing in the function definitions within **card.cpp** should help you add the missing statements to **card.cpp**
- Once **Card** is complete, then work on **BlackCard** and **RedCard** and so on.

There is one function, Description, used by the Print function in main.cpp that must exhibit polymorphic operation.

You may need to utilize **Constructor Initializers** in the derived classes.

<Project02 Debugging Assistance >

Your program must be fully commented (including variable and parameter declarations, function descriptions, descriptions of logical blocks of code, etc.) in order to receive debugging assistance from the instructor and teaching assistants.

<Important Project02 Submission Instructions>

- (1) Follow all submission instructions.
- (2) Below is a list of the eight files you must submit. It is very important that you use the correct file names: [lowercase letters, no spaces, no underscores] in order for your program to compile and execute.

card.h	(modify and submit the provided file)
card.cpp	(modify and submit the provided file)
blackcard.cpp	(write and submit this file using the information from blackcard.h)
redcard.cpp	(write and submit this file using the information from redcard.h)
club.cpp	(write and submit this file using the information from club.h)
diamond.cpp	(write and submit this file using the information from diamond.h)
heart.cpp	(write and submit this file using the information from heart.h)
spade.cpp	(write and submit this file using the information from spade.h)

A COMPLETE SUBMISSION is one that includes all eight files listed above as attachments to the same submission. All other submissions will be considered to be incomplete.

INCOMPLETE SUBMISSIONS will receive zero credit (0 points) on this assignment.

DO NOT USE A FILE COMPRESSION PROGRAM (.zip, .tar, etc. are not allowed)

Failure to satisfy this requirement will result in zero credit (0 points) on this assignment.

- (3) **Do not modify or submit the file named main.cpp!!!**

Failure to satisfy this requirement will result in zero credit (0 points) on this assignment.

- (4) **All output to the monitor (stdout) will be performed by the code provided in main.cpp**

NOTE: Your project submission will be automatically graded so it is EXTREMELY IMPORTANT that you READ and FOLLOW the project directions.

Submit only the specified files to Canvas.

Submissions by email receive ZERO CREDIT (0 points)!!!

Submissions that do not compile receive ZERO CREDIT (0 points)!!!