CS 200 – Intro to Programming

Assignment 5 [Fall 2018]

Release Date: **Monday** 26th November 2018, 5:00 PM

Due Date: ***Sunday*** 2nd December 2018, 11:55 PM

Please keep in mind the following guidelines:

- Do not share your program code with anyone.
- Do not copy code from the internet.
- If you receive any assistance, mention the part of code in which you received assistance.
- You must be able to explain any part of your submitted code.
- All submissions are subjected to automated plagiarism detection.

Submission:

You have to submit all the .cpp files containing source code. Zip all .cpp files into one file named as <your8DigitRollNumber> .zip and submit the zip file.

Note: Define a class interface separately and its methods separately. Do not write inline code. Your code needs to compile and run to get marks

Task 1: Rock Paper Scissors (50 marks)

A. Implement a class called **Tool** (5)
   a. It should have an integer field called ***strength*** and a *char* field called ***type***. You may make them either private or protected.
   b. It should also contain the constructor(s), destructor as well as getters and setters for **strength** and **type** and a virtual function **fight**.

B. Create 3 more classes called **Rock**, **Paper**, and **Scissor**, which inherit from class **Tool**. (10)
   a. The default constructor will initialize to strength 1
   b. Each of these classes will need a parametric constructor which will take in an int that is used to initialize the **strength** field.
   c. The constructor should also initialize the **type** field using 'r' for Rock, 'p' for Paper, and 's' for Scissors.

C. These classes will also need to define the function ***bool fight(Tool)*** that compares their strengths in the following way: (10)
   a. Rock's strength is doubled (temporarily) when fighting scissors, but halved (temporarily) when fighting paper.
   b. In the same way, paper has the advantage against rock, and scissors against paper.
   c. The function ***bool fight()*** returns true if the original class wins in strength and false otherwise.

D. Write a test function to create objects (dynamic) of derived classes using the pointers of base class and verify the results. (10)

   You can set the strengths as follows:

Scissors:	5
Paper:	7
Rock:	15

E. Create a class called ***RPSPlay***, which allows a human to play the game against the computer. (15)
   a. Your RPSPlay shall have two ***Tool ****, one for human and the other for the computer respectively, because it is not known at the start of the game which tool will be selected.
   b. The RPSPlay game shall also have three int fields to keep track of the number of **human_wins**, **computer_wins**, and **ties**.

Note:

- You may also include any extra auxiliary functions and/or fields in any of these classes.
- Provide overloaded assignment operator for each class.
- You may need to use random number generator to implement some of the functionality in E.

Task 2: Appointment Book (25)

A. Implement a base class ***Appointment*** and derived classes **Onetime**, **Daily**, **Weekly**, and **Monthly**. (10 = 2+2+2+2+2)
   a. An appointment has a description (for example, “see the dentist”) and a date and time.
   b. Create a separate class Date to store Date and Time.
   c. Write getters and setters for the base class.
   d. Write appropriate constructors and destructors for all the classes.

B. Write a virtual function **occurs_on(int year, int month, int day)** that checks whether the appointment occurs on that date. (10)
   a. For example, for a monthly appointment, you must check whether the day of the month matches.
   b. Write a test function to fill an array of Appointment* with a mixture of appointments.

Have the user enter a date and print out all appointments that happen on that date. (5)