**EN 001203 Computer Programming**

**Mid-Term Exam**

**Faculty of Engineering, Khon Kaen University**

**Academic Year 2563 Semester 2**

**19 February 2021, 13:00 – 16:00**

**Instructions:**

1. There are 9 problems each worth 60 points. Full scores require every problem solved.
2. This is a closed book exam.   
   **\* No other reading materials are allowed.**
3. **Network communication is allowed only for   
   (a) submission of the answers to the designated system and   
   (b) consulting reference, i.e.,** [www.cppreference.com](http://www.cppreference.com) and [www.cplusplus.com](http://www.cplusplus.com)**.  
   \* Personal communication, social media, file sharing (in other systems), or internet searching is NOT allowed.**
4. Name each file as follows:

\* Name your submission program by the corresponding problem: **Px.cpp**

For example, P1.cpp for problem 1. P2.cpp for problem 2, and so on.

1. Put all submission files to a single tar file.
   1. Use a proper compression utility (with a proper setting)
2. Submit the program through the designated system.

====================================================

The designated system is **autolab.en.kku.ac.th**

Version Policy:

A student can submit as many as please. However, version(s) beyond version 5 will be penalized 5% for each excessive version.

The system will allow submission until 16:15, but a late submission after 16:00 will be penalized 10%.

====================================================

**P1.** Output.

Write a program to print out “TRUST” on the screen. The program takes no input.

Output example:

|  |
| --- |
| TRUST |

**P2.** Input.

Write a program to take 2 inputs from a user: name and age. Then display the information as follows:

Output example, when a user puts in Namtarn and 25 respectively.

|  |
| --- |
| Name: **Namtarn**  Age: **25**  25 years ago, Namtarn was born. |

The **bold font** indicates input. Fonts are used only to emphasize a user input. The program is not expected to produce any of these font effects.

[*Hint: declare*

string myString;

*To get a string variable to store text data.*]

**P3.** Expression.

Write a program that gets 2 **integers** and stores into x and y respectively. Then find a real number value of z = 3x / 4y

Output example: when inputting 6 and 8 respectively via keyboard

|  |
| --- |
| Enter 2 integers: **6 8**  z = 0.5625 |

**P4.** Flow rate.   
Write a program to calculate a flow rate (liter/second) from a volume of liquid flowing through a pipe and being accumulated for one minute. The program asks a user for a volume of liquid (in liter) accumulated for one minute.

Have the program interact exactly like what shown in the example. Volume is an input, which has to be obtained from a user.

[*Hint: flow rate = volume/time and 1 minute = 60 seconds.*]

Output example: when inputting the volume as 2.4

|  |
| --- |
| Volume: ***2.4***  Flow: 0.04 |

**P5.** Child height.  
Write a program to take the height of a child in cm. Then determine whether a child can play a ride. If the child is at least 120 cm high, the child can play. If not, the child cannot play.

Output example 1, when a user puts in 150.

|  |
| --- |
| Height: **150**  Can play |

Output example 2, when a user puts in 110.

|  |
| --- |
| Height: **110**  Cannot play |

**P6.** Monopole.

Write a program to calculate a length of a monopole antenna (L), given a target radio wave frequency (f). The program asks a user for a radio wave frequency (in Hz) and calculates a length (in meter) of a quarter-wave monopole, according to:

L = 𝜆 / 4

where L is a length (in meter) of a monopole antenna and 𝜆 is a wavelength (in meter).

Note that c = f ∙ 𝜆 where f is a frequency (in Hz); 𝜆 is a wavelength (in meter); and c is a light speed, using meter/second for c.

Have the program interact exactly like what shown in the example. Frequency (f) is an input, which has to be obtained from a user.

Output example: when inputting the frequency as 2400000

|  |
| --- |
| Frequency: ***2400000***  Length: 31.25 |

**P7.** Body temperature.Write a program to determine the fever level and output a message according to the following table, when a user inputs a body temperature.

|  |  |
| --- | --- |
| **Temperature** | **Message** |
| < 35.0 | Hypo |
| 35.0 - 37.5 | Normal |
| 37.6 - 40.0 | Fever |
| > 40.0 | Doctor! |

Output example 1, when a user puts in 36.8.

|  |
| --- |
| Tmp: **36.8**  Normal |

Output example 2, when a user puts in 32.

|  |
| --- |
| Tmp: **32**  Hypo |

Output example 3, when a user puts in 50.

|  |
| --- |
| Tmp: **50**  Doctor! |

Note: provided examples do not include all cases.

**P8.** Exam score.  
Write a program to calculate your exam score, according to the same policy as this exam as follows.

* If a student submits 5 versions or less, the student will not be penalized.
* Each version exceeding 5 versions will be penalized 5%.

For example,

If a student submits 4 versions, there is no penalty.

If a student submits 5 versions, there is no penalty.

If a student submits 6 versions, there is 5% off penalty. If the total Autolab score is 200, the actual score will be 200 - 5% = 190 points.

If a student submits 7 versions, there is 10% off penalty. If the total Autolab score is 200, the actual score will be 200 – 10% = 180 points.

If a student submits 8 versions, there is 15% off penalty. If the total Autolab score is 200, the actual score will be 200 – 15% = 170 points.

And so on.

Take a raw score and a number of versions as inputs from a user.

Output example 1, when a user puts in 200 and 4 respectively.

|  |
| --- |
| Raw score: **200**  Version: **4**  Actual score: 200 |

Output example 2, when a user puts in 540 and 8 respectively.

|  |
| --- |
| Raw score: **540**  Version: **8**  Actual score: 459 |

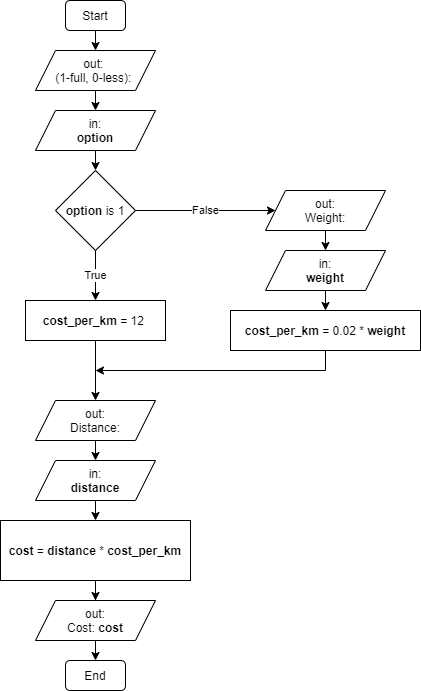
Output example 3, when a user puts in 600 and 10 respectively.

|  |
| --- |
| Raw score: **600**  Version: **10**  Actual score: 450 |

Note: provided examples do not include all cases.

**P9.** Freight cost.

Write a program to calculate a freight cost. The cost is charged 12 baht/km for a full truckload or 0.02 baht/km per kg for a less-than-truckload freight. See the flowchart below (Figure 1) for clarification.



*Figure 1. Freight-cost flowchart. Variables are shown in a* ***bold font****.*

The program asks a user if he/she wants to take a full truckload (1) or a less-than-truckload (0). If the user takes a less-than-truckload, ask for a weight of the package. Either case the program asks for a distance, then computes a freight cost and reports.

Have the program interact exactly like what shown in the examples. Option, weight, and distance are input, which have to be obtained from a user.

**!Caution!**: The flowchart shows variables **option**, **weight**, **cost\_per\_km**, **distance**, and **cost**. It is recommended to declare **option** as an integer and other variables as floating-point numbers.

Output example: when choosing a full truckload with distance of 440 km

|  |
| --- |
| (1-full, 0-less): ***1***  Distance: ***440***  Cost: 5280 |

Note: choosing 1 (in the first line) represents choosing a full truckload.

Output example: when choosing a less-than truckload with weight of 50 kg and distance of 440 km

|  |
| --- |
| (1-full, 0-less): ***0***  Weight: ***50***  Distance: ***440***  Cost: 440 |

Note: choosing 0 (in the first line) represents choosing a less-than truckload.

Output example: when choosing a less-than truckload with weight of 100 kg and distance of 110 km

|  |
| --- |
| (1-full, 0-less): ***0***  Weight: ***100***  Distance: ***110***  Cost: 220 |