Worth: 10% **Due:** Before 10:00pm on Thursday 8 October 2015.

Your submission must be a PDF file named a1.pdf and it must be handed-in using the MarkUs system. You are to create the PDF file using a document preparation system (*e.g.* Word or LaTeX). Only hand in the PDF file - do not hand in other files (*e.g.* a .doc file or a .tex file). (If you expect to have a difficult time with the typesetting, speak with Tom.)

For questions where you are asked to write a program, you must include your program and any output in your submission.

See the Syllabus for the course policy on late assignment submissions.

---

1. (a) In IEEE arithmetic how many double precision numbers are there between any two adjacent nonzero single precision numbers? Justify your response.

   (b) What is the largest integer p such that all integers in the interval $[-p, p]$ are exactly representable in IEEE double precision arithmetic? What is the corresponding p for IEEE single precision arithmetic? Justify your responses.

2. (a) The following C program:

```
#include <stdio.h>

int main (void)
{
    double a, b;
    for (a = 0.1; a <= 0.5; a += 0.1)
        printf("a = %e\n", a);
    for (b = 1.1; b <= 1.5; b += 0.1)
        printf("b = %e\n", b);

    return 0;
}
```

   outputs five values of `a` and four values of `b`. Explain why the two loops do not produce the same number of lines of output.

   (Ask for assistance if you do not know C. The programming language is not the issue here.)

   (b) Suppose that you are writing a program to run on a computer that supports IEEE floating point arithmetic, have declared and initialize a floating point variable named `x`, and need to determine 10% of the value stored in `x`. You need to decide between using the expression `0.1 * x` or `x / 10.0`. Which of these two expressions would you recommend using if accuracy of the final result is your biggest concern? Explain.

3. You know that $\cos(\ 2m\pi\ ) = 1$ for every integer m. Using a programming language of your choice that uses IEEE fixed precision floating-point arithmetic, write a program that calculates `cos( 2*m*pi )` for `m` $= 10.0^{\mathtt{k}}$, `k` $= 0, 1, 2, \ldots, 20$, where `pi` is either your language's builtin approximation to π or `4.0*arctan(1.0)`. Present the results of your calculations in a formatted table, with column headings for the values of `k`, `2*m*pi`, and `cos( 2*m*pi )`. Use a "scientific notation" to present the values of `2*m*pi` and `cos( 2*m*pi )`, giving 15 significant figures for both `2*m*pi` and `cos( 2*m*pi )`.

   Carefully explain why you do not always compute `cos(2*m*pi)` ≈ 1.0 and why the approximations change significantly for larger values of `k` .

4. Consider the problem of evaluating the function $f(x)$ for any $x \in [-\pi/2, +\pi/2]$, where

$$f(x) = \begin{cases} \dfrac{1 - \cos(x)}{\sin(x)}, & x \in [-\pi/2, 0) \cup (0, +\pi/2] \\ 0, & x = 0. \end{cases}$$

   (a) Show why one can expect to be able to write a computer program that uses floating-point arithmetic to accurately evaluate $f(x)$ at any $x \in [-\pi/2, +\pi/2]$.

   (b) Derive an accurate algorithm for evaluating $f(x)$ and explain why you think that accurate results will be computed. Make use of standard math library functions like `sin()` and `cos()` when expressing your algorithm.

If you have questions or encounter any difficulties, post a description of the issue on the Piazza course forum.