Workshop 1 – Sample Solutions

1) Playing with escape sequences

Output:

```
Hello
World
Hello\nWorld
Hello\
World
Hello World
HellWorld
World
```

Explanation:

- `printf(“Hello\nWorld\n”);`
  First prints **Hello** then ‘\n’ i.e. new line (cursor moves to a new line) and then prints **World** and then cursor moves to new line by ‘\n’.

- `printf(“Hello\\nWorld\n”);`
  First prints **Hello** then the compiler reads \\, which means to print single backslash so compiler prints backslash (\), then the compiler reads **nWorld** and print its and then cursor moves to new line by ‘\n’.

- `printf(“Hello\\\nWorld\n”);`
  First prints **Hello** then the compiler reads \\, which means to print single backslash so compiler prints backslash (\), then the compiler reads \n which means new line so compiler move the cursor to the new line and then prints **World** and then cursor moves to new line by ‘\n’.

- `printf(“Hello\tWorld\n”);`
  First prints **Hello** then ‘\t’ i.e. horizontal tab, the cursor moves the tab space and then prints **World** and then cursor moves to new line by ‘\n’.

- `printf(“Hello\bWorld\n”);`
  First prints **Hello**, then ‘\b’ means backspace which move the cursor one position to the left of its current position, so letter o is erased and last result is **HellWorld** and then cursor moves to new line by ‘\n’.

- `printf(“Hello\rWorld\n”);`
- First prints **Hello**, then ‘\r’ means carriage return which moves the cursor to the beginning of its current line from where it prints **World** and then cursor moves to new line by ‘\n’.

2) Simple calculations using arithmetic operators

```
#include int main()

{

float circ, area, radius = 0; // float data type!

const float PI = 3.141592; // use of a constant

printf("\nPlease enter the radius of the circle in meter: ");

scanf("%f", &radius);

if (radius < 0)

      printf("The radius must be a positive number! Please try
      again.\n”);

else

{

      circ = PI * radius * 2; // 2*PI*r

      area = PI * radius * radius; // PI*r^2

      // use the %.x specifier to indicate precision for float

     printf("The circumference of the circle is %.3fm.\n", circ);

     printf("The area size of the circle is %.3fm.\n", area);

}

}
```

3) Recognising and avoiding mistakes

```
#include <stido.h>
int main()     {
     float x,y; //initializing two variables
     Printf(“Enter 1st number: “);
     scanf(“%f”,x); //taking first input
     printf(“Enter 2nd number: “)
     scanf(“%d”,&y); //taking second input
     sum = x+y; //computing sum of two inputs
     prinft(“The sum of %f and %f is: /n %f,”x,y,sum);
     return 0;
```

Capital P in printf

& missing

%f is used with float

; missing

Variable sum is not initialized

/ used instead of \

} for closing main() missing

4) Relational operators within a while loop

```
#include <stdio.h>
int main() {
int number = 0;
while (number != 42)
{
      printf("\nPlease enter a number : ");
      scanf("%d", &number);
      if (number >= 0)
            printf("%d is a positive number.\n", number);
      else
            printf("%d is a negative number.\n", number);
}
printf("You found the answer to life, the universe, and
everything!");
return 0;
}
```