Eli Griffiths and Cristian Aguirre
Checked by groups 3 and 6

Struct - Worksheet

```
struct Product
{
        string description; // Product description
        int partNum; // Part number
        double cost; // Product cost
};
```

1. Write a definition for an array of 100 Product structures. Do not initialize the array.

```
Product products[100];
```

2. Write a loop that will step through the entire array you defined in Question 1, setting all the product descriptions to an empty string, all part numbers to zero, and all costs to zero.

```
for (int i = 0; i < 100; ++i)
{
   products[i] = {};
}
```

3. Write the statements that will store the following data in the first element of the array you defined in Question 1:
   Description: Claw hammer
   Part Number: 547
   Part Cost: $8.29

```
products[0] = {
        .description = "Claw hammer",
        .partNum = 547,
        .cost = 8.29
};
```

4. Write a loop that will display the contents of the entire array you created in Question 1.

```
for (int i = 0; i < 100; ++i) {
    cout << "---------------\n";
    cout << "Desc  : " << products[i].description;
    cout << "Part #: " << products[i].partNum;
    cout << "Cost  : " << products[i].cost;
}
```

5. Write a structure declaration named Measurement, with the following members:
   miles, an integer
   meters, a long integer

```
struct Measurement
{
    int miles;
    long meters;
};
```

6. Write a structure declaration named Destination, with the following members:
   city, a string object
   distance, a Measurement structure (declared in Question 5)
   Also define a variable of this structure type.

```
struct Destination
{
    string city;
    Measurement distance;
};

Destination dest = {
    .city = "Pasadena",
    .distance = {
        .miles = 60,
        .meters = 96238
    }
};
```

7. Write statements that store the following data in the variable you defined in Question 6:
   City: Tupelo
   Miles: 375
   Meters: 603,375

```
Destination dest = {
    .city = "Tupelo",
    .distance = {
        .miles = 375,
        .meters = 603375
    }
};
```

Assume the following structure declaration exists for Questions 8-10:

```
struct Rectangle
{
    int length;
    int width;
};
```

8. Write a function that accepts a Rectangle structure as its argument and displays the structures contents on the screen.

```
void display_rect(Rectangle rect)
{
    cout << "Length: " << rect.length << endl;
    cout << "Width : " << rect.width << endl;
}
```

9. Write a function that uses a Rectangle structure reference variable as its parameter and stores the users input in the structures members.

```
void get_rect(Rectangle& rect)
{
    cout << "Length: ";
    cin >> rect.length;
    cout << "Width : ";
    cin >> rect.width;
}
```

10. Write a function that returns a Rectangle structure. The function should store the users input in the members of the structure before returning it.

```
Rectangle get_rect()
{
    Rectangle rect;
    cout << "Length: ";
    cin >> rect.length;
    cout << "Width : ";
    cin >> rect.width;
    return rect;
}
```