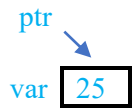
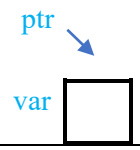
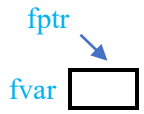
SECJ 1013 PROGRAMMING TECHNIQUE 1
ASSIGNMENT 3

GROUP MEMBER'S (MATRIC NO):
Lee Jia Yee A24CS0260
Leong Jia Ling A24CS0104

SECTION: 02

1) State whether the following declarations are valid or invalid. Give reasons for the invalid declarations and draw memory layout for the valid declarations. (7 marks)

i.	`int var = 25;` `int *ptr = &var;`	Valid. 
ii.	`int var = 30;` `int* ptr = var;`	Invalid. Assigning an `int` directly to a pointer is not allowed, it must be assigned the address of a variable.
iii.	`int var, *ptr;` `ptr = &var;`	Valid. 
iv.	`float fvar;` `int *ptr = &fvar;`	Invalid. Initializing a pointer with the address variable of different data type is not allowed. When assigning an address `&fvar` to a pointer `*ptr`, the different datatype are incompatible.
v.	`float fvar, *fptr = &fvar;`	Valid. 
vi.	`int *ptr = &var;` `int var = 25;`	Invalid. The address of variable `var` is used before declaration. The variable `var` must be declared before address variable `var` can be assigned to a pointer.

vii.	`double* dptr1, dptr2;`
	`double dvar = 25.2;`
	`dptr1 = &dvar;`
	`dptr2 = &dvar;`
		Invalid.
		`dptr2` is declared as a variable of `double` datatype, but not a pointer. So, `dptr2 = &dvar` is invalid because a `double` datatype cannot store an address.

2) Determine the output and draw a memory layout (or memory allocation) of the pointers and variables for code segment below. Note: Draw a memory layout that represents C++ statement line by line. (7 marks)

```
int  x = 10, y = 20, z = 30;
int *ptr;

cout << x << " "  << y << " " << z << endl;
ptr = &x;
*ptr *= 10;
ptr = &y;
*ptr *= 4;
ptr = &z;
*ptr *= 2;

cout << x << " "  << y << " " << z << endl;
```

	Memory Layout
`int x=10, y=20, z=30;` `int *ptr;`	x [10] y [20] z [30]
`ptr=&x;` `*ptr*=10;`	ptr → x [100] y [20] z [30]
`ptr=&y;` `*ptr*=4;`	ptr → x [100] y [80] z [30]
`ptr=&z;` `*ptr*=2;`	ptr → x [100] y [80] z [60]
Output: 10 20 30 100 80 60	ptr → x [100] y [80] z [60]

3) Write two statements to free dynamically allocated array and double which are declared as follows: (2 marks)

```
int *iPtr = new int [100];
double *dPtr = new double;
```

```
delete[] iPtr;
delete dPtr;
```

4) Starting address of the following array named iVar is 0xFEC07.

[0]	[1]	[2]	[3]
2	5	8	6

iVar

What is the output that will be displayed based on the following statements? (4 marks)

i.	cout << iVar;	0xFEC07
ii.	cout << iVar [0];	2
iii.	cout << *iVar;	2
iv.	cout << *(iVar + 2);	8

5) Write a structure declaration to hold the following data (6 marks)

i. About a flight reservation: passenger name, age, reservation code, departure location, destination, flight number, departure time, arrival time, cost and payment status.

```
struct FlightReservation
{
    string passengerName, departureLocation, destination, flightNumber;
    int age, reservationCode;
    float departureTime, arrivalTime, cost;
    bool paymentStatus;
};
```

ii. About saving account: account number, account balance, interest rate, total deposit and total withdraw.

```
struct SavingAccount
{
    int accountNumber;
    double accountBalance, interestRate, totalDeposit, totalWithdraw;
};
```

iii. About PT1 assessments: student's name, test 1, assignment, quiz, lab exercise, final exam, course work mark, total mark and grade.

```
struct PT1Assessments
{
    string studentName;
    double test1, assignment, quiz, labExercise, finalExam, courseworkMark, totalMark;
    char grade;
};
```

6) A car salesman keeps the information of each model of car he sells. The example of information for 3 cars' models is as in Table 2. Write C++ statement for the following task. (10 marks)

Model	Engine capacity	Price
Waja	1.6	60000
Wira	1.5	50000
MyVi	1.3	45000

i. Define a structure for storing the above information named Car.

```
struct Car
{
    string model;
    double engineCapacity;
    int price;
};
```

ii. Declare a variable called myCar and initialized it with some values of your choice. Display information on myCar.

```
Car myCar= {"Waja", 1.6, 60000};
cout << "Information of my car" << endl;
cout << "Model: " << myCar.model << endl;
cout << "Engine capacity: " <<  myCar.engineCapacity << " cc"<< endl;
cout << "Price: RM " << myCar.price << endl;
```

iii. Declare another variable called mySecondCar and assign values to it using assignment statements. Display information on mySecondCar.

```
Car mySecondCar;
mySecondCar.model= "Wira";
mySecondCar.engineCapacity=1.5;
mySecondCar.price=50000;
cout << "Information of my second car" << endl;
cout << "Model: " << mySecondCar.model << endl;
cout << "Engine capacity: " <<  mySecondCar.engineCapacity << " cc"<< endl;
cout << "Price: RM " << mySecondCar.price << endl;
```

iv. Print the total of price paid for myCar and mySecondCar.

```
int totalPrice= myCar.price+ mySecondCar.price;
cout << "The total of price paid for myCar and mySecondCar is: RM " << totalPrice << endl;
```

v. Copy the values and information of mySecondCar into myCar and display current information on myCar.

```
myCar= mySecondCar;
cout << "Current information of myCar after copying from mySecondCar: " << endl;
cout << "Model: " << myCar.model << endl;
cout << "Engine Capacity: " << myCar.engineCapacity << " cc" << endl;
cout << "Price: RM " << myCar.price << endl;
```

7) Write the code segment for each of the following tasks: (8 marks)

a) Declare a structure type:

i. named `Salary`, with the following members:
`basic` : a double value
`allowances` : a double value

```
struct Salary
{
    double basic;
    double allowances;
};
```

ii. named `Employee`, with the following members:
`name` : a string value
`id` : an integer value
`salary` : a `Salary` structure variable

```
struct Employee
{
    string name;
    int id;
    Salary salary;
};
```

iii. Declare a variable of structure type `Employee` named `myEmp`.

```
Employee myEmp;
```

b) By using the variables and structure declaration in (a), define a function named `displayEmp`. It should accept an `Employee` structure variable as its argument and not return a value. The function should display the contents of the variable onto the screen based on figure below. *Notes: Assuming the data for struct members was already assigned.

```
Sample output:
Name: Azira
Id: 8902
Basic salary: RM 4500
Allowances: RM 500
```

```
void displayEmp (const Employee &emp)
{
    cout << "Sample output: " << endl;
    cout << "Name: " << emp.name << endl;
    cout << "Id: " << emp.id << endl;
    cout << "Basic salary: RM " << emp.salary.basic << endl;
    cout << "Allowances: RM " << emp.salary.allowances << endl;
}

int main()
{
    Employee myEmp;
    myEmp.name= "Azira";
    myEmp.id=8902;
    myEmp.salary.basic=4500.0;
    myEmp.salary.allowances=500.0;

    displayEmp (myEmp);

    return 0;
}
```